AD-A099 331 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 20/4
PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLU--ETC(U)
NOV 80 J MULLEN, F K GOODWIN, M F DILLENIUS F33615-76-C-3077
UNCLASSIFIED NEAR-TR-210-VOL-3 AFWAL-TR-80-3032-VOL-3 NL

1 of 3
AD
A099331

AD A099331

AFWAL-TR-80-3032
VOLUME III

PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLUDING FUSELAGE AND STORES OF NONCIRCULAR CROSS SECTION. VOLUME III - APPENDICES A AND B, DETAILS OF PROGRAM I

Joseph Mullen, Jr.
Frederick K. Goodwin
Marnix F. E. Dillenius
Nielsen Engineering & Research, Inc.
Mountain View, California 94043

November 1980

TECHNICAL REPORT AFWAL-TR-80-3032, VOLUME III
FINAL REPORT FOR PERIOD JUNE 1975 - FEBRUARY 1980

Approved for Public Release, Distribution Unlimited

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Calvin L. Dyer
CALVIN L. DYER
Project Engineer

R.O. Anderson
RONALD O. ANDERSON, Chief
Control Dynamics Branch

FOR THE COMMANDER

Robert C. Ettinger
ROBERT C. ETTINGER, Col, USAF, Chief
Flight Control Division

"If your address has changed, if you wish to be remived from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIGC, W-PAFB, OH 45433 to help us maintain a current mailing list."

(19) TR-80-3032-VOL-3

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWAL-TR-80-3032, Vol. III	2. GOVT ACCESSION NO. AD-A099	3. RECIPIENT'S CATALOG NUMBER 332
4. TITLE (and Subtitle) (6) PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLUDING FUSELAGE AND STORES OF NONCIRCULAR CROSS SECTION. Volume III. Appendices A and B, Details of Program I.		5. TYPE OF REPORT & PERIOD COVERED (9) Final Report. June 1975 — February 1980
		6. PERFORMING ORG. REPORT NUMBER (14) NEAR-TR-210-VOL-3
7. AUTHOR(s) (10) Joseph Mullen, Jr. Frederick K. Goodwin Marnix F. E. Dillenius		8. CONTRACT OR GRANT NUMBER(s) (15) F33615-76-C-3077
9. PERFORMING ORGANIZATION NAME AND ADDRESS Nielsen Engineering & Research, Inc. 510 Clyde Avenue Mountain View, CA 94043 (12) 207		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) Project 2403 Task 240305 (17) 05 Work Unit 240309
11. CONTROLLING OFFICE NAME AND ADDRESS Flight Dynamics Laboratory Air Force Wright Aeronautical Laboratories Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433 (11)		12. REPORT DATE November 1980
		13. NUMBER OF PAGES 210
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aerodynamic Loads
Aerodynamic Interference
External Stores
Flow Fields
Store Separation
Supersonic Flow

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Detailed instructions are presented for using a computer program which calculates the six-degree-of-freedom trajectories of external stores which are separated from fighter-bomber type aircraft flying at supersonic speeds. Multiple circular or elliptic store configurations may be handled. Parent aircraft configurations may consist of a circular or arbitrary cross section fuselage with ramp external compression inlets, and a wing, pylon, and rack. The program uses linear potential-flow theory to model the wing and pylon loading and thickness. Three-dimensional line sources and doublets are used

DD FORM 1 JAN 73 1473

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

to model circular fuselages and stores. The noncircular fuselage and elliptic store surfaces are modeled with constant source panels. Nonlinear corrections are made to the wing, fuselage, rack, store and fuselage inlet models to simulate shocks. The program also calculates the trajectory of the store as it separates from the aircraft. This report describes the program, presents instructions for preparing input for the program, describes the output from the program, and presents a sample case. The program represents an extension of an earlier program restricted to circular bodies at supersonic speeds written by the present authors and described in AFFDL-TR-76-41.

This volume presents the detailed descriptions of the calculations performed in each of the subroutines in Program I. Also included are the descriptions of each of the variables passed between routines.

A

FOREWORD

This report entitled "Prediction of Supersonic Store Separation Characteristics Including Fuselage and Stores of Noncircular Cross Section," describes a combined theoretical-experimental program directed toward developing a computer program for predicting the trajectory of an external store separated from an aircraft flying at supersonic speed. It represents an extension of previous work covered in AFFDL-TR-76-41 to include more realistic modeling of fuselage shapes including noncircular cross sections and ramp type engine air inlets, and to include modeling store shapes with elliptic cross section with multiple sets of arbitrarily oriented fins. Volume I, "Theoretical Methods and Comparisons with Experiment", describes the theoretical approach and presents extensive comparisons with experimental data. Volume II, "Users Manual for the Computer Program", presents detailed instructions on the use of the computer program with emphasis on preparation of input data and interpretation of output. This volume, Volume III, "Appendices A and B, Details of Program I", provides additional descriptions of the individual subroutines and program variables passed between modules in the first of two programs. Volume IV, "Appendices C and D, Details of Program II", provides additional descriptions of the individual subroutines and program variables passed between modules in the second program.

This work was carried out by Nielsen Engineering & Research, Inc. (NEAR), 510 Clyde Avenue, Mountain View, California 94043, under Contract No. F33615-76-C-3077. The contract was initiated under Project 2403, Task 240305, of the Air Force Flight Dynamics Laboratory. The Air Force Project Engineer on the contract was Calvin L. Dyer, AFWAL/FIGC. The report number assigned by Nielsen Engineering & Research, Inc. is NEAR TR 210.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLUDING FUSELAGE AND STORES OF NONCIRCULAR CROSS SECTION

Volume III - Appendices A and B, Details of Program I

SUMMARY

The purpose of this volume is to provide the additional details of the parts of the first program that would be useful to the programmer or engineer interested in understanding the calculations and computer code herein. The information included here as Appendices A and B describes the function and operations performed by each routine, a description of data transferred between subroutines and a listing of Program I itself.

Appendix A provides a detailed description of the operations and flow of calculations of each of the individual routines in Program I. Included are a description of the flow of the calculations, including flow charts of some routines, a description of any program arguments, and a program listing.

Appendix B provides a description of all variables passed between routines in common blocks. A listing of each common block with a description of each variable, array, or index in the common is provided. A special section is provided for the multiple uses of blank common as well as a cross reference chart of routine versus common block usage.

APPENDIX A
DETAILS OF PROGRAM I SUBROUTINES

A-1 Introduction

The purpose of this appendix is to provide more detailed information on the calculations performed in the first program which was described in Section 3 of Volume II. A listing of Program I is presented in Figure A-1 and a general flow chart in Figure A-2. This appendix will present the details of the flow of the calculations for each routine, a description of the variables in the argument list, individual detailed flow charts, and a summary of the functions of each routine. The program consists of a main program and 59 subroutines. The main program will be described and then the subroutines will be described in alphabetical order. The subroutines and their functions are listed in Table A-1. Flow charts of some of the individual routines are shown in Figures A-3 through A-18. Refer to Volume II of this report for the list of symbols used in this text.

TABLE A-1. SUBROUTINES USED IN PROGRAM I

Subroutine Name	Function
LDCALC	main program to organize calculation flow
ASECTN	calculates cross sectional area of an arbitrarily paneled body
BDYGEN	calculates line sources and doublets to give a required body shape and angle of attack
BLYOUT	lays out fuselage constant u-velocity panels for circular body
BLYOT2	lays out noncircular fuselage constant u-velocity panels

BODPAN	organizes the calculation of the revised axial spacing for the noncircular fuselage source panels
BODPN2	performs the revision of the axial spacing on the noncircular bodies and computes the body panel geometry
BODVEL	organizes the looping through source panels to compute panel influence coefficients
BSHOCK	computes the nonlinear shock wave shape emanating from the nose of a noncircular body
CONFIG	reads and prints the input geometrical description of the external shape of noncircular fuselage or elliptic stores
DOUBLT	calculates the strength of a linear line doublet
DPCOEF	calculates coefficient matrix of the set of equations to be solved for the wing-pylon-fuselage constant u-velocity panel strengths
DPRHS	calculates right-hand side of the set of equations to be solved for the wing-pylon-fuselage constant u-velocity panel strengths
FLDVEL	organizes the calculation of the u,v,w velocity components at the field points XFP,YFP,ZFP
FLDVL2	computes the three components of velocity induced at field points by the source panels of a given body ring
FRSTRT	saves or restores required program information for source panel model to restart configuration analysis
FUSEIO	reads and prints fuselage data and calls BDYGEN to calculate line source and doublet distributions and SHKSHP to compute nonlinear shock for circular bodies
GEOM	organizes the reading and printing of the noncircular body input and the calculation of the source panel geometry arrays

INLSHK	computes the nonlinear shock shape emanating from a ramp type inlet
INLTST	determines whether a source panel is an inlet panel
INLXYZ	scans the x,y,z coordinates of the inlet panel corners to determine inlet geometric parameters
INVER1	solves a system of simultaneous linear algebraic equations
IOREAD	performs an unformatted read from external file, IO
IOWRIT	performs an unformatted write onto the external file, IO
NETCLC	calculates the net corner strengths for a given set of wing, pylon, or fuselage constant u-velocity panels
NEWRAD	organizes the revision of the noncircular body meridian line spacing for the source panels
NEWRD2	computes the revised noncircular body meridional line spacing for a given body segment
NULYT	selects a subset of constant u-velocity and wing and pylon thickness panel layout data and calculates and saves the net strengths associated with each panel corner
PANEL	calculates the direction cosines of the normal vector, the centroid, area and inclination angles of an arbitrary quadrilateral panel
PANVEL	transforms field points into local panel coordinates and calls SORPAN for the calculation of the panel influence coefficient
PAS001	performs the L*U decomposition of a positive definite matrix
PAS002	solves the system of equations [L*U]*X = B
PLYOUT	reads and prints pylon data and lays out constant u-velocity panels on the pylon

PYTHIN	reads pylon thickness data
RACKIO	reads and prints the rack data and calculates the rack source and doublet distributions
SHAPE	calculates the radius and surface slope at a point on a body from input polynomials
SHKSHP	calculates the nonlinear shape of the shock wave shape produced by an axisymmetric body at zero angle of attack
SMARCH	solves for the source panel strengths in supersonic flow using a ring by ring marching technique
SOLVE	computes the source panel boundary condition for a noncircular body in the presence of a free stream angle of attack and calls SMARCH to compute the panel strengths
SORPAN	computes the three velocity components induced at a specified control point by a constant source panel inclined to the free stream
SOURCE	calculates the strength of a linear line source
STORIO	reads and prints store data and computes circular body line source and doublet distributions and shock shapes or elliptic store source panel strengths and shock shape
SWNGIN	reads and prints wing constant u-velocity panel data and twist and camber data, if any
THKLYT	lays out wing and pylon thickness panels
THKOUT	prints wing and pylon thickness panel input data
VELBD1	calculates influence functions due to a fuselage constant u-velocity panel
VELCAL	calculates velocities at a field point due to fuselage or store line sources and doublets
VELCMP	organizes and calls for the calculation of the aerodynamic influence coefficient matrices for the body source panels

VELO1	calculates the influence of a semi-infinite triangle associated with a constant u-velocity panel
VELOT1	calculates the influence of a semi-infinite triangle associated with a thickness source panel
VELPP1	calculates influence functions due to a pylon constant u-velocity panel
VELPT1	calculates velocities at a field point due to pylon thickness source panels
VELWP1	calculates influence functions due to a wing constant u-velocity panel
VELWT1	calculates velocities at a field point due to wing thickness source panels
WDYBDY	reads and prints noncircular fuselage data and organizes the calls for the body source panel layout, source strengths and nonlinear shock shape
WITHIN	reads wing thickness data
WLYOUT	lays out constant u-velocity panels on wing
WRFILE	writes the dataset on TAPE12 which contains the information required to continue the supersonic store separation computations in Program II
YZBIP	organizes the scan of the source panel geometry to define the cross section of the noncircular fuselage interference shell
YZBIP2	interpolates in the panel geometry of a segment to define the y-z values of the noncircular interference shell

A-2 Program LDCALC

The flow chart which was presented in Figure 2 of Volume II details the basic flow of the calculations in program LDCALC and will not be expanded in this appendix. A written summary of the calculations to be made in the first program is presented. A listing of the program is presented in Figures A-1(a) and A-1(b) of this report.

LDCALC organizes the reading and printing of input and the calculation of the strengths of the singularities modeling the individual parent aircraft and store components. The constants, flow conditions, and component options are first defined. The source strength solutions are then defined for the circular fuselage in FUSEIO or for the noncircular fuselage in WDYBDY. The wing u-velocity panel data are read in SWNGIN and the panel layout is defined in WLYOUT. The wing thickness data are read in WITHIN. PLYOUT and PYTHIN perform similar functions for the pylon. The thickness distribution is printed from THKOUT and the thickness panel layout for wing and pylon occurs in THKLYT. The u-velocity panel layout for the circular fuselage is generated in BLYOUT and in BLYOT2 for the noncircular fuselage. A source strength model may also be generated for the rack in RACKIO. The source strength determination for either circular or elliptic store bodies are then computed in STORIO. The right hand side and influence coefficients for the u-velocity panels are computed in DPRHS and DPCOEF, respectively. The panel solutions are solved for from the set of simultaneous equations in INVER1. A summary of the panel control points and solutions is printed. The modified layout based on strengths summed at panel corners is then generated in NULYT. The necessary arrays and variables are lastly saved on TAPE12 in WRFILE for restarting the problem in the second program.

Subroutine references

BLYOUT, BLYOT2, DPCOEF, DPRHS, FUSEIO, NULYT, PLYOUT, PYTHIN, RACKIO, STORIO, SWNGIN, THKLYT, THKOUT, WDYBDY, WITHIN, WLYOUT, WRFILE.

A-3 Subroutine ASECTN

Subroutine ASECTN computes the cross sectional area of an arbitrarily paneled body. It is used to compute the area enclosed by arbitrary Y and Z values at each station in the external specification of the body shape. It is called from CONFIG. The equation used to define the area is

$$AREA = \sum_{i=2}^{NRAD} \frac{1}{2}(Y_i \cdot Z_{i-1} - Y_{i-1} \cdot Z_i) \qquad \text{(A-1)}$$

If the body is symmetric the sum is computed for the half body and doubled. A listing of the routine is presented in Figure A-1(b) of this report. A description of the parameters in the argument list follows:

Y,Z	arrays of section corner points
NRAD	number of points around section
AREA	area of cross section

Called by

CONFIG

A-4 Subroutine BDYGEN

Subroutine BDYGEN calculates the line source and doublet strengths using control points on the surface of the fuselage and store. The method used is described in Appendix I of Reference 1. A listing of the subroutine is presented in Figure A-1(c) and a flow chart in Figure A-3 of this report. The

coordinate system associated with the subroutine is shown in the sketches of Appendix I of Reference 1.

At the beginning of the subroutine a test is performed to determine if, at the base of the body, the radial distance to the Mach cone emanating from the body nose is less than the maximum radius of the body. If so, an error message is printed out (see Section 3.5 of Volume II) and the program stops.

Next, as the flow chart indicates, N is set equal to NXBODY - 1 and the body axis is divided into N segments of equal length. The x locations of the body definition points, XBODY(J), are determined at these equally spaced axis points and, inside a short DO loop, subroutine SHAPE is called to calculate the radius of the body, RBODY, and the surface slope, RPBODY, at each body definition point. Subroutine SHAPE requires that the shape definition quantities be made dimensionless by body length. Subroutine BDYGEN accounts for this before and after the calls to subroutine SHAPE. Next, the control points are located effectively midway between the body definition points. Subroutine SHAPE calculates the body radius, RF, and the surface slope, DRDX, at each control point. Finally, the axis points, TX, which are the origins of the conical line sources and doublets, are determined.

The next sections of the subroutine are devoted to revising the layout of the body definition points and control points and the origins of the line singularities if the first control point lies outside the Mach cone originating at the body nose. An iteration is performed to determine the intersection of the cone with the body surface. Once this point is found the body definition points are redistributed over the remainder of the body. The procedure described in the previous paragraph is used

to redefine the control points and the origins of the line singularities.

The remainder of the subroutine calculates the source and doublet strengths at the control points. Subroutine SOURCE is called and the source strength at the first control point is calculated using Equation (I-14) and at the remaining control points using Equation (I-17) of Reference 1. One should note that the Ith source strength, T(I), is the constant K_{I-1} in the algebraic notation of the equations. Similarly, using subroutine DOUBLT, the doublet strengths are calculated using Equation (I-27) for the first control point and Equation (I-28) of Reference 1 for the remaining control points. Here, the Ith doublet strength, TC(I), equals $K_{d,I-1}$.

After calculating the source and doublet strengths, the subroutine prints the body definition point data, the singularity origins, and the singularity strengths and sums the source and doublet strengths. A description of the parameters in the argument list follow:

NXBODY	number of body definition points
RADIUS	maximum radius of body
BODYL	length of actual body
NSEG	number of polynomials used to specify body shape
XEND	array containing x locations of end points of polynomial sections specifying body shape, made dimensionless by body length
SCOEF	array containing coefficients of polynomials used to specify body shape

T	array containing source strengths; T(J) is the quantity K_{J-1} calculated by Equations (I-14) and (I-17) of Reference 1
TC	array containing doublet strengths; TC(J) is the quantity $K_{d,J-1}$ calculated by Equations (I-27) and (I-28) of Reference 1
TX	array containing x locations of origins of conical line sources and doublets; positive, measured from tip of nose
ALPHA	angle measured between the body centerline and the free-stream direction
RPBODY(I)	slope of body surface at Ith body definition point
LBODY	BODYL*XEND(NSEG)
SUMK	sum of the source strengths
SUMKD	sum of the doublet strengths

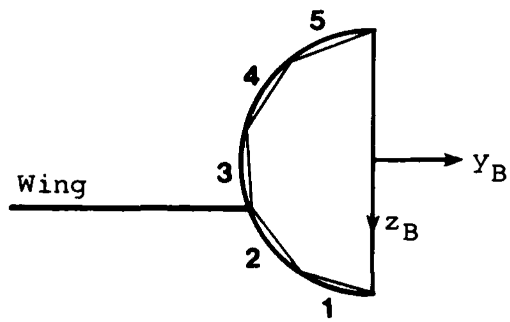
Subroutine references
DOUBLT, SHAPE, SOURCE

Called by
FUSEIO, RACKIO, STORIO

A-5 Subroutine BLYOUT

Subroutine BLYOUT calculates quantities needed to define the constant u-velocity panels on the circular fuselage and to locate the panel control points. The corner and control point coordinates calculated for the body panels are stored in arrays in locations which follow the same quantities calculated for the wing and pylon constant u-velocity panels. All coordinates are in the wing coordinate system which is shown in Figure 6 of Volume II. A listing of the subroutine is presented in Figure A-1(d) and a flow chart in Figure A-4.

The fuselage interference panels are laid out on the left-half of the fuselage surface, as shown in Figure 3 of Volume II. The panel numbering convention employed in a typical ring containing five panels, three above the wing and two below, is shown in the following sketch.

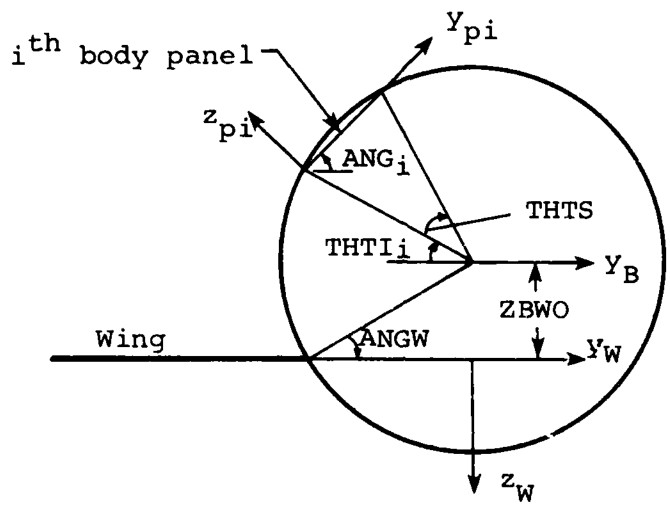


At the beginning of the subroutine, quantities DX, NBD, NBIP, and ANGW, associated with the geometry of the entire panel layout, are calculated. Next, the variable ANGW is tested to determine if the wing is tangent to the fuselage, either at the top or at the bottom. As the flow chart shows, this tangency condition determines the initialization procedure performed before the main loop calculations.

The main part of the subroutine is a double DO loop with the outer loop index running over the rows above or below the wing and the inner loop index over the NCWB panels in a lengthwise

row. If the wing is not tangent, this double loop is performed first for the panels below the wing, then repeated for the panels above the wing.

For a given lengthwise row the y,z coordinates of the panel corners and the control points are first obtained in the panel coordinate system. The following sketch shows the coordinate system and certain quantities associated with a panel located above the wing.



Within the inner DO loop the coordinates are transformed into the wing coordinate system and stored in appropriate array locations. Polar angle THTI and quantities SNT2 and CST2, necessary for panel-wing transformations, are saved in arrays for subsequent use by other routines. The corner and control point x_w coordinates are calculated and stored; no transformations are necessary for these quantities.

The convention employed in labeling the corner coordinates for the fuselage interference panels is that the right side of the panel is located in clockwise rotation from the left side, when looking forward.

Called by

LDCALC

A-6 Subroutine BLYOT2

Subroutine BLYOT2 computes the u-velocity panel corner points, control points, and transformation angles for the noncircular fuselage body shape. The section geometry is obtained from YZBIP at the user specified body x-station in the arbitrary fuselage geometry description. The section generated is held constant in cross section in the axial direction. No logic has been provided to force panel edges at the wing-body interface to match. The initial paneling of the body by the user must be performed so as to guarantee the matching of the panel edges with the wing root chord. A listing of the routine is presented in Figure A-1(e) of this report.

BLYOT2 performs two loops when generating the tables of panel properties. The outer DO loop steps through the number of panels around the circumference to generate the local section properties as shown in Figure 18 of Volume II. The inner DO loop increments the indices axially to repeat the identical section geometry for the remaining downstream panels on the body interference shell. The panel properties are laid out starting at the bottom for the left hand side. All quantities defined maintain the same definitions and conventions used in BLYOUT for the circular fuselage.

Called by
LDCALC

A-7 Subroutine BODPAN

Subroutine BODPAN organizes the calculation of the revised axial spacing for noncircular fuselage source panels. The starting locations in blank common of all arrays used to define the geometric properties of the panels are computed. The routine

loops on the number of body segments used to specify the body shape. Subroutine BODPN2 is called for each of NFUS segments to calculate the revised axial spacing on the body and the body panel geometry. If more than one segment exists, the revised meridional spacing geometry, YB and ZB, are read from TAPE8. At the end all geometric arrays in blank common are saved on TAPE7. The optional print of source panel control point coordinates, inclination angles and areas are performed here. A listing of this routine is found in Figure A-1(e) of this report.

See the discussion in Appendix B, Section B-3 regarding item 2 in Figure A-2 for the equivalent dimension of blank common at this point. The starting location indices defined in BODPAN and saved in labeled common DIMENS are

```
IXPT=1
IYPT=IXPT+NBODY
IZPT=IYPT+NBODY
ITH=IZPT+NBODY
IDEL=ITH+NBODY
IAR=IDEL+NBODY
IXC(IFU)=IAR+KFORX(1)+...+KFORX(IFU)   IFU=1,NFUS
IYC(IFU)=IAR+KX+KFORX(1)·KRADX(1)+...+KFORX(IFU)·KRADX(IFU)
IZC(IFU)=IYC(IFU)+KXKR
```

Subroutine references

IOREAD, BODPN2, IOWRIT

Called by

GEOM

A-8 Subroutine BODPN2

Subroutine BODPN2 performs the revision of the axial spacing on noncircular bodies and computes the body panel geometry of a

single segment. The x, y, and z coordinates of the cross sections of a segment are passed through the argument list as arrays. A listing of the routine is presented in Figure A-1(f) of this report.

The body panel geometry is established by a linear interpolation along body meridian lines of the y and z coordinates at the new axial stations. The interpolation is started with the first ring of panels in a segment and continued until the last ring of panels on the last segment is reached. The corner point coordinates, the control point coordinates, the inclination angles, and area are calculated for each panel in sequence. The calculation of the latter quantities are performed in routine PANEL from corner point information.

The panel control point coordinates, the panel dihedral angle θ, the panel incidence angle δ, the corner point coordinates and the panel areas are returned through the argument list to BODPAN where they are stored in blank common. Finally if the print option IPRT(2)$\neq$0, the corner point coordinates are written on the output file.

Argument list

JP	panel number
XB(M)	x station of external geometry for Mth section in segment
XJ(J)	x station of revised axial spacing for Jth section in segment
YB(M,K), ZB(M,K)	y and z of external geometry for Mth section in segment and Kth meridional station
XYZ(JP,1), XYZ(JP,2), XYZ(JP,3)	XPT, YPT, and ZPT arrays of the x, y, and z coordinates of the control points of JPth panel
XYZ(JP,4)	dihedral angle THET of the JPth panel

XYZ(JP,5)	incidence angle DELTA of the JPth panel
XYZ(JP,6)	area of JPth panel
XC(J)	corner points of panels at Jth axial station
YC(J,K), ZC(J,K)	y and z corner points of panels at Jth axial station and Kth meridional angle
NFUSOR	number of external geometry stations on segment
KRAD	number of meridan lines on segment
KFUSOR	number of panel revised axial stations
NBODY	total number of body panels
IPRT	array of print controls

Subroutine references

PANEL

Called by

BODPAN

A-9 Subroutine BODVEL

Subroutine BODVEL organizes the looping through source panels to define corner geometry and compute panel influence coefficients. BODVEL loops through the number of source panels on each ring in the given body segment to set up the local panel corner coordinates. A listing of this routine is presented in Figure A-1(f) of this report.

The influence coefficients for each panel on all remaining panels is set up and computed in PANVEL. The influence coefficients are then generated for one ring of body panels on another ring. For each such block of coefficients, the influence normal to the panel and each of the three component velocities, u, v, and w, are saved. The block of normal coefficients is saved on TAPE9. If the coefficients correspond to a diagonal block in the total coefficient matrix (i.e. the influence of a ring on itself) the

L*U triangularization of the block is performed by PAS001 for later calculation use. In supersonic flow it is assumed that aft rings of panels have no influence on upstream panels. All blocks of coefficients above the diagonal blocks are thus assumed to be zero and not computed.

The descriptions of the variables in the argument list follow

XPT,YPT,ZPT	coordinates of panel control points
THET	array of panel inclination angles at control points
DELTA	array of panel incidence angles
AN	temporary array to contain the normal velocities to control point I
UB(1,IJ), UB(2,IJ), UB(3,IJ)	arrays of velocity components, u,v,w, at control points in body coordinate system
XC	x coordinates of leading and trailing edges of panel rings of segment
YC,ZC	y and z coordinates of body panel corner points of segment
KFUSOR	number of axial stations defining segment
IFU	segment index
IXZSYM	body XZ plane of symmetry indicator

Subroutine references

IOWRIT, PANVEL, PAS001

Called by

VELCMP

A-10 Subroutine BSHOCK

Subroutine BSHOCK computes the nonlinear shock wave shape emanating from the store or fuselage nose produced by an arbitrary body at angle of attack along a given meridian, PHIS(J). Three options are available depending on the value of NSHOCK (input items 14 or 61) for the choice of where the meridional traverses are made. Single traverses, evenly distributed traverses, and user specified locations may be used to define polar shape of the shock wave. A listing of the routine is presented in Figure A-1(g) of this report.

The methods used to generate the nonlinear position of the shock wave along a single traverse are developed in Section 4.1.2 of Reference 3. Use is made of the work of Reference 2. The axial scan for the maximum radial velocity behind the linear shock wave is computed for a fixed interval equal to half the distance from the nose to the body shoulder. This approximate interval must be used to account for the saw tooth behavior of the velocity field. The three-dimensional effects associated with angle of attack are accounted for by two means. In computing the local radial velocities along a traverse, the perturbation components must be rotated into wind axes to avoid including angle of attack components directly in the radial velocities as follows

$$u_w = u_s \cos\alpha_c - v_s \sin\alpha_c \sin\phi_R + w_s \sin\alpha_c \cos\phi_R + 1.0 \qquad \text{(A-2)}$$

$$v_w = v_s \cos\phi_R + w_s \sin\phi_R \qquad \text{(A-3)}$$

$$w_w = -u_s \sin\alpha_c - v_s \cos\alpha_c \cdot \sin\phi_R + w_s \cos\alpha_c \cdot \cos\phi_R \qquad \text{(A-4)}$$

and second, the assumption the Mach cone rotates rigidly about the body with angle of attack is modified in accordance with reference 2 as follows.

$$\theta = \theta_s + \alpha_c(1 - \varepsilon_\alpha) \cos\phi \qquad \text{(A-5)}$$

where θ is the shock angle propagating from the nose, θ_s is the nose limited shock angle at zero angle of attack, α_c is the included angle of attack, ε_α is Mach number and nose cone angle dependent correction from Reference 2, and ϕ is the polar angle around the cone of the traverse. Tables of RSHK and XSHK for each of the polar traverses are the primary output of this routine.

This routine is used in two ways. For the fuselage, the shock shape is computed at the angle of attack of the body. The traverses are computed over the entire field of interest and include the corrections for angle of attack. For the stores, the shock shape is computed at zero angle of attack for only the first quadrant. The remaining shape is derived from symmetry. The angle of attack correction of Equation (A-5) is then applied to the zero angle of attack shape for the local conditions of each of the elliptic stores.

The descriptions of the parameters of the argument list are as follows:

RINIT	minimum radius at which shock shape search is initiated
PHISHK	for NSHOCK=0, angle at which shock shape is generated for NSHOCK<0, maximum meridional angle to which shock shape is calculated on even spacing
THSHK	limiting shock wave angle at nose measured from x-axis, θ_s

Subroutine references

FLDVEL

Called by

WDYBDY, STORIO

A-11 Subroutine CONFIG

Subroutine CONFIG is used to input the geometrical description of the external shape of the noncircular fuselage or elliptic stores. A listing of the routine is presented in Figure A-1(i) of this report. The routine first reads the configuration reference area from the input file if $J0 \neq 0$, otherwise the reference area is set equal to unity.

If $J2 \neq 0$, the body external geometry data is read from the input file according to the specified body option for each of NFUS body segments. For arbitrary cross-sections, the y and z ordinates of the body segment are read. For circular or elliptic sections, the particular combination of body area, radius, semi-major or minor axes, or elliptic ratio requested may be read.

In addition, the area at each axial station is computed. The maximum cross-section area is also saved. The description of the single argument is as follows:

SFUS	array to hold the y and z ordinates of the arbitrary cross-section data for J2=1

Subroutine references

ASECTN

Called by

GEOM

A-12 Subroutine DOUBLT

Subroutine DOUBLT calculates coefficients used in the determination of the line doublet strengths. They occur as terms in Equations (I-27) and (I-28) of Reference 1. The relation of the coefficients to perturbation velocities, $u_{B,d}/V_\infty$ and $v_{B,d}/V_\infty$, induced by a number of line doublets distributed along the body centerline is shown in the first two of Equation (I-30) in which the coefficients occur as the multipliers of $K_{d,n} \cos\theta$. A listing of the subroutine is presented in Figure A-1(j) of this report. Section I-2.2 of Reference 1 should be referred to for further details concerning the doublet strength calculations.

For a specified control point, x_B, r_B, and singularity origin ξ, the subroutine calculates quantities U and V according to the following equations.

$$U = \beta\sqrt{\left(\frac{x_B - \xi}{\beta r_B}\right)^2 - 1} \qquad \text{(A-6)}$$

$$V = -\frac{\beta^2}{2}\left[\cosh^{-1}\left(\frac{x_B - \xi}{\beta r_B}\right) + \left(\frac{x_B - \xi}{\beta r_B}\right)\sqrt{\left(\frac{x_B - \xi}{\beta r_B}\right)^2 - 1}\right] \qquad \text{(A-7)}$$

At the beginning of the subroutine a test is performed to determine if the control point is ahead of the Mach cone from the doublet origin. If so, U and V are set to zero and control is returned to the calling program.

The following table of definitions contains most of the variable names used in the subroutine:

BETA	$\beta = \sqrt{M_\infty^2 - 1}$
RFIELD	r_B; radius of body at control point
TX	ξ; location on body axis of doublet origin, positive measured from tip of nose
U	coefficient defined by Equation A-6
V	coefficient defined by Equation A-7
XFIELD	x_B; x location of control point, positive measured from tip of nose

Called by

BDYGEN

A-13 Subroutine DPCOEF

Subroutine DPCOEF calculates the coefficient matrix of the set of simultaneous boundary condition equations which are to be solved for the constant u-velocity panel singularity strengths. A listing of the subroutine is presented in Figure A-1(k), and a flow chart in Figure A-5 of this report.

The elements of the matrix are the aerodynamic influence coefficients described in Section 3.3.4 of Reference 3. They occur indirectly in the summation terms in the left-hand side of Equations (24), (25), and (26) of that reference. The actual coefficients which the subroutine calculates, FVN(ν,n), are related to the summation terms through the panel strengths, u_{+n}/V_∞, by the following equations in which n is the index of the influencing panel and ν is the control point index:

$$\frac{w_{w_{\nu,n}}}{V_\infty}\cos\phi_\nu - \frac{v_{w_{\nu,n}}}{V_\infty}\sin\phi_\nu = \frac{u_{+n}}{\pi V_\infty}\,\mathrm{FVN}(\nu,n) \qquad \nu = 1,2,\ldots\mathrm{NPANLS}$$

$$\frac{v_{w_{\nu,n}}}{V_\infty} = \frac{u_{+n}}{\pi V_\infty}\,\mathrm{FVN}(\nu,n) \qquad \nu = \mathrm{NIP},\mathrm{NIP}+1,\ldots\mathrm{N2}$$

$$\frac{v_{N_{\nu,n}}}{V_\infty} = \frac{u_{+n}}{\pi V_\infty}\,\mathrm{FVN}(\nu,n) \qquad \nu = \mathrm{N2P},\mathrm{N2P}+1,\ldots\mathrm{NPTOT}$$

for n = 1,2,...,NPTOT.

The subroutine consists of three double DO loops. The first loop uses subroutine VELWP1 to calculate the influence of the wing panels at the wing, pylon, and fuselage control points. The second loop is bypassed if there is no pylon, NPY=0. If a pylon is present, the influence coefficients for the pylon panels are calculated by means of subroutine VELPP1. The third double loop is bypassed if there is no fuselage, NFU=0. Otherwise, the influence coefficients for the fuselage panels are calculated using subroutine VELBD1.

Within each outer loop are three inner loops in series, which fix the control point location on the wing, pylon, or fuselage, respectively. On the flow chart, only the first occurrence of the first inner loop is shown in detail, which includes the call to subroutine VELWP1. The remaining inner loops have the same logical structure and are not shown.

If the control point is located on a wing or a fuselage panel, the influence coefficient is the component normal to the panel surface at the control point. Thus, the wing coefficients are rotated through the panel dihedral angle using quantities previously calculated in subroutine WLYOUT. The fuselage coefficients are rotated through the panel orientation angle using quantities calculated in subroutine BLYOUT.

Subroutine references

VELBD1, VELPP1, VELWP1

Called by

LDCALC

A-14 Subroutine DPRHS

Subroutine DPRHS calculates the right-hand side vector of the set of simultaneous boundary condition equations which are to be solved in order to determine the constant u-velocity panel singularity strengths. The boundary conditions are specified in Equations (24), (25), and (26) of Reference 3. A listing of the subroutine is presented in Figure A-1(k) and a flow chart in Figure A-6 of this report.

The major portion of the subroutine is devoted to evaluating the externally induced perturbation velocities, $u_{w_{i,\nu}}/V_\infty$, $v_{w_{i,\nu}}/V_\infty$, and $w_{w_{i,\nu}}/V_\infty$, at all of the wing, pylon, and fuselage control points.

The first section calculates velocities induced at wing and pylon control points by the circular fuselage line sources, sinks, and line doublets if a circular fuselage is present (NFU=1). A control point is located in the fuselage coordinate system and subroutine VELCAL is called to calculate the velocities at this point. These velocities are summed in the UEI, VEI, and WEI arrays.

The next section calculates velocities induced at wing and pylon control points by the noncircular fuselage source panels if a noncircular fuselage is present (NFU=2). Data for the fuselage are first restored to blank common from TAPE10 through FRSTRT and indices defining the beginning of the various arrays are defined. All of the wing and pylon control points are then located in the fuselage coordinate system. The velocities at these points are calculated by calling subroutine FLDVEL.

Next, the perturbation velocities induced by the wing and pylon thickness source panels are calculated by means of subroutines VELWT1 and VELPT1. The effects included are wing

source panels on wing, pylon, and fuselage control points and pylon source panels on wing and fuselage control points. If a pylon is not present, NPY=0, or the fuselage is not present, NFU=0, calculations involving the pylon or the fuselage are omitted.

The last three loops in the subroutine calculates the right-hand sides of the equations. If the pylon is located below the fuselage centerline, CENTER=TRUE, VEI is set to zero for all pylon control points.

It should be noted that if the control point is located on the wing or on the fuselage, dihedral effects are included in calculating the velocity normal to the panel surface.

Subroutine references

FLDVEL, FRSTRT, VELCAL, VEIPT1, VELWT1

Called by

LDCALC

A-15 Subroutine FLDVEL

Subroutine FLDVEL is used to organize the computation of the u, v, w velocity components at the field points XFP, YFP, ZFP. This routine initializes all velocities to zero and computes the locations of the required geometric and strength arrays in blank common for the fuselage or store at hand. If the configuration models the fuselage with inlets, variable INLET is initialized to TRUE to indicate presence of inlet panels. It then performs the looping for the number of segments and the number of rings in each segment to sum the component velocity contributions at each field point. A listing of the routine is presented in Figure A-1(m) of this report.

The descriptions of the parameters in the argument list follow:

XFP,YFP,ZFP	arrays of coordinates of field points in coordinate system of body panels at which component velocities are computed
SVN	temporary array of length NFLD used to test for zero influence at field point
LSKP	temporary logical array of length NFLD used to test for zero influence of ring of panels at field point
U,V,W	arrays of component velocities computed for influence of body at field points
NFLD	number of field points
ID	array containing the same variables in the same order as in common DIMENS
IG	index specifying IGth strength solution, GB, to be used

Subroutine references

FLDVL2

Called by

BSHOCK, DPRHS, INLSHK

A-16 Subroutine FLDVL2

Subroutine FLDVL2 computes the three components of velocity induced at specified field points by the source panels of a given ring of body panels. The field point is assumed to have no incidence or inclination relative to the flow. A listing of the routine is presented in Figure A-1(m) of this report.

The routine first initializes the field point inclination and incidence transformations to zero. For each of the influencing

panels on a ring the control point coordinates, inclination and incidence angles and corner points in the local panel system are defined. For each panel β_ℓ is initialized to β_o. For panels used to model inlet openings β_ℓ is set to the minimum of β_o and BTINLT. Further, if an inlet panel slope exceeds BTINLT, β_ℓ is set to 0.99/tanδ. For each of the field points the influence of the panel on the field point is computed, multiplied times the panel strength and summed.

To minimize computational time FLDVL2 tests whether each ring of panels contributes to the total velocity at a given field point. For each point the following sum is made for each panel on the ring

$$SVN(I) = UB^2 + VB^2 + WB^2$$

where UB, VB, WB are the influence coefficients of a panel at the Ith field point. If the net influence for a complete ring SVN is equal to zero the logical variable LSKP(I) for that field point is set .TRUE. and all further calculations at that point are suppressed. This is based on the assumption that at supersonic speeds once a point is ahead of the Mach waves from a ring, the influence of that ring and all subsequent rings will vanish.

The descriptions of the parameters in the argument list follow:

XPT,YPT,ZPT	arrays of the coordinates of the source panel control points for the entire body
THET	array of inclination angles for panels
DELTA	array of incidence angles for panels
GB	array of panel strengths
XFP,YFP,ZFP	arrays of coordinates of field points
U,V,W	arrays of orthogonal velocity components at panels in x, y, and z directions

SVN	temporary array used to test influence of ring on point
LSKP	temporary array used as logical indicator of no further influence at point
NFLD	number of field points
XC,YC,ZC	arrays of coordinates of panel corners for segment
KFUSOR	number of axial station bounding panels in segment
IXZSYM	body symmetry indicator
JR	number of panels in ring J
JG	starting offset index of first panel in ring
L	index of trailing x-station in segment

Subroutine references

PANVEL, INLTST

Called by

FLDVEL, IMAGEV

A-17 Subroutine FRSTRT

Subroutine FRSTRT is used to save or restore required program information for the source paneling method to restart a configuration analysis. The routine is set up to allow storage of the information from more than one configuration on the same file. A listing of the routine is presented in Figure A-1(n) of this report. Six options are available based on the information to be stored or retrieved and where the information is to reside. They and the functions they perform are

KODE	DESCRIPTION
1	saves control integers and arrays in blank common
2	restores control integers and arrays in blank common

3 saves above information and AIC, U, V, and W velocity matrices

4 restores above information and AIC, U, V, and W velocity matrices

5 restores control integers and arrays stored under either option 1 or 3 into starting core location in blank common. All arrays are copied into blank common rather than into original labeled commons

6 restores information in manner of KODE=5 and reads past velocity matrices stored under option 3 to position file at next record

The information saved under KODE=1 consists of all the variables in labeled commons DIMENS, PARAM, BOPTNS, BGEOM, HEAD, and BSHOCK, and the first NTAP7 variables in blank common. The later set are also saved on TAPE7. If the configuration contains an inlet, the variables in common BINLET and BINSHK are also saved. In copying the velocity matrices under KODE=2, the arrays are first copied from TAPE8 or TAPE9 into temporary arrays in blank common and then onto the output file IO. The last two options are used in Program II to stack multiple configurations end-to-end in blank common. Provision has been made for only one configuration with an inlet. Variables from commons BINLET and BINSHK are copied back into those arrays under all KODE options. During this data retrieval three indices used to locate the control variable arrays are saved.

IDO (=LASTA+1)	first location in blank common of variables contained in labeled commons DIMENS, PARAM, BOPTNS, BGEOM, and TITLE in order
ISKO (=IDO+571)	first location in blank common of variables contained in labeled common BSHOCK

IAO (=ISKO+240)	first location in blank common of geometric and strength arrays previously saved on TAPE7

The descriptions of the parameters in the argument list are:

IO	external file unit number onto which the data is stored or retrieved in unformatted form
KODE	optional index selecting information to be read or written to file IO; see above descriptions
LASTA	last location in blank common currently defined

Subroutine references
IOREAD, IOWRIT

Called by
WDYBDY, STORIO, DPRHS, WRFILE

A-18 Subroutine *FUSEIO*

Subroutine FUSEIO reads and prints the input data which describe the circular fuselage (NFU=1) and calculates the line source and doublet distributions as described in Appendix I of Reference 1. A listing of the subroutine is presented in Figure A-1(o) and a flow chart in Figure A-7 of this report.

The subroutine first reads in and prints input items 5, 6, 7, and 8 which consist of the fuselage length, the maximum radius, and the polynomials specifying the fuselage shape. Next, the data used to lay out the body interference panels, input items 9 and 10, are read and printed. Subroutine BDYGEN is then called for the purpose of calculating the source and doublet distributions. The fuselage shoulder location is next found and finally

subroutine SHKSHP is called to calculate the nose shock wave shape.

Subroutine references

BDYGEN, SHKSHP

Called by

LDCALC

A-19 Subroutine GEOM

Subroutine GEOM organizes the reading of the noncircular body input and the calculation of the source panel geometry arrays. This routine is used to input both the noncircular fuselage and the elliptic store panel geometry. A listing of the routine is presented in Figure A-1(p) of this report.

The specification of the input describing the body paneling is split in two phases by GEOM. In the first phase the control parameters and the geometric arrays describing the external shape of the body are read. The external geometry arrays are read by a call to CONFIG. This geometry defines the external shape independently of the paneling used. It may optionally be used to specify panel corner coordinates. In the second phase the control parameters, additional arrays and inlet data required are read which define the subdivision of the external shape into panel coordinates. The routine NEWRAD is called to perform the interpolation necessary to redefine the meridional spacing of the panels. The routine BODPAN is called to perform the interpolations for new axial spacings of panels and to calculate control point coordinate and inclination information.

This routine also computes the number of panels to be generated and the allocation of array space in blank common to be used. The total number of axial stations, meridional angles used to

define panel corners and the total length of blank common required are compared against dimensioned values as check on input problem size.

Subroutine references

BODPAN, CONFIG, NEWRAD

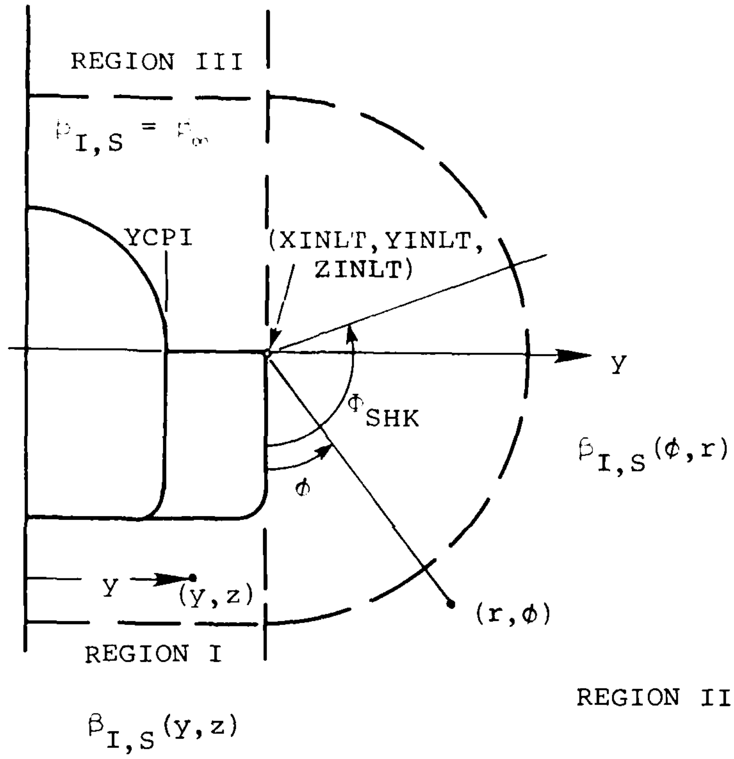
Called by

WDYBDY, STORIO

A-20 Subroutine INLSHK

Subroutine INLSHK computes the nonlinear shock wave shape emanating from a ramp type inlet shown in Figure 12 of Volume II. The table of shock locations is produced in a manner analygous to that used in BSHOCK to generate the shock shape about the body nose with the exceptions that only the influence of the modified panels across the inlet are felt and the shock propagates from the leading edge of the ramp. The detailed explanation of the inlet shock is presented in Section 4.4.2 of Reference 3. A listing of the routine is presented in Figure A-1(r) of this report.

The generation of the traverses used to locate the first influence and integrate the nonlinear shape uses the same model and equations detailed for BSHOCK. Additional assumptions are made about the body in generating the traverses regarding the shape of the ramp inlet. Field points aft of the inlet are assumed to lie in the three regions relative to XINLT, YINLT, ZINLT as shown in the following sketch.



In region I inside and below the inlet, traverses are carried out parallel to the XZ plane with subsequent interpolation in the y-direction to find the first inlet shock influence, $\beta_{I,S}$. In region II, outboard of the inlet, traverses are calculated along radial lines from (XINLT,YINLT,ZINLT) from $\phi = 0^o$ to $\phi = \Phi_{SHK}$. The last traverse at $\phi = 180^o$ is assumed such that $\beta_{I,S} = \beta_\infty$. In region III inboard and above Φ_{SHK} for the inlet, the flow is assumed equal to free stream for which $\beta_{I,S}$ is set to β_∞.

NIS shock traverses are generated alternating first in Region I and then in Region II to define the inlet shock location. The shocks propagate from a starting point at (XINLT, y, ZINLT) along a line at the angle ϕ from the vertical plane. The first shock is positioned at y=YINLT and ϕ=0^o. The second traverse propagates

from y=YINLT at ϕ=PHISHK. The third propagates from y=0 at ϕ=0°. The fourth is at y=YINLT and ϕ=45° and the fifth is at y=YCPI and ϕ=0°. Additional traverses are laid out by subdividing region I between y=0 and y=YCPI and region II between ϕ=0 and ϕ=PHISHK with even increments.

The following are differences in the calculative procedure for the inlet and the nose shock shape. In scanning in the axial direction at a given Y,Z location, all values are referenced to XINLT, YINLT(IRAD), and ZINLT. In computing the influence at field points the strengths of all panels except those designated as open or blocked inlet panels and additional adjacent panels on the inlet aft of the opening are set equal to zero. In integrating the traverses, no angle equivalent to the nose limiting shock angle, θ_s, is used. When estimating the shock location below the inlet in region I in the presence of partial blockage (mass flow ratios less than one), the shock is assumed to follow BTINLT down the inlet face to the first blocked panel. A normal shock is assumed to exist moving the shock vertically down to the lower lip of the cowl. For the first step, thereafter, in the radial direction, the $\beta_{I,S}$ propagating from the inlet is set equal to the first computed value. The only restriction is that it be positive.

The description of the parameter in the argument list follows:

PHISHK maximum meridional angle about YINLT,ZINLT to which shock shape is calculated on even spacing

Subroutine references
FLDVEL, INLTST

Called by
WDYBDY

A-21 Function INLTST

Logical function INLTST determines whether the panel index in the argument is for an inlet panel. The value of the function is set to TRUE if the index is for an inlet panel and FALSE if it is not. In addition the logical variable OPEN is set to indicate whether the inlet panel is blocked or unblocked to flow. Three conditions are tested for. If no inlet panels exist INLTST and OPEN are set false. If an inlet panel exists the index I is compared to the table of possible inlet panel numbers, JINLT. If I is equal to one of the inlet panel numbers, INLTST is set true. If I is not an inlet panel number, INLTST is set false. If I falls in the subset of JINLT of unblocked panels, OPEN is true; otherwise OPEN is set false. A listing of this routine is presented in Figure A-1(t) of this report.

The description of the parameters in the argument list follows.

I	panel number index to be compared with table of possible inlet panel numbers, JINLT
OPEN	logical variable indicating whether an inlet panel is open or blocked. OPEN is TRUE if panel allows unblocked flow through panel

Called by

BODVEL, FLDVL2, PANEL, SOLVE

A-22 Subroutine INLXYZ

Subroutine INLXYZ scans the X,Y,Z coordinates of the inlet panel corners to find the coordinates to be used to define inlet

leading edge, lower cowl lip, and first axial station containing blocked panels. Nine values are computed. They correspond to the most outboard leading edge of the inlet panels, XINLT,YINLT,ZINLT, the most aft outboard trailing edge of the inlet panels, XINLTE, YINLTE,ZINLTE, the innermost y station of the inlet, YCPI, and the first axial station of blocked inlet panels, XCLOSD. If no inlet blockage exists,this value is set to inlet trailing edge XINLTE. In addition, the inlet mass flow ratio is computed from the ratio of exposed open frontal area of inlet panels, A_{open}, to total inlet frontal area, A_{total}. The mass flow ratio RVIVO is set equal to A_{open}/A_{total}. This routine is called once for each inlet panel. A listing of the routine is presented in Figure A-1(t) of this report.

The descriptions of the parameters in the argument list follow.

XIN,YIN,ZIN	coordinates of corner of inlet panel to be checked
FAREA	frontal area of panel
OPEN	logical variable indicating whether an inlet panel is open or blocked. OPEN is TRUE if panel allows unblocked flow through panel

Called by

PANEL

A-23 Subroutine INVER1

Subroutine INVER1 solves the system of simultaneous linear algebraic equations.

$$[A]\overline{X} = \overline{B}$$

This routine performs pivot searching during the solution of the general matrix, A. The right hand side is passed into INVER1 as

the N+1'st column in the matrix. The solution, X, also returns in that location. A listing of this routine is presented in Figure A-1(u) of this report. The routine is currently limited to 200 equations by internal dimensions. The descriptions of the parameters in the argument list follow.

A	coefficients of linear system of equations in first N columns; columns N+1 through N+NSYS contain multiple right-hand sides, B, on input and solutions, X, on return
NSYS	number of right hand sides
N	actual number of equations
NMAX	first dimension of A
MMAX	second dimension of A

Called by
LDCALC

A-24 Subroutine IOREAD

Subroutine IOREAD performs an unformatted read from external file, IO. NA consecutive elements of array, A, are read sequentially. This routine is used to specify a common interface to external files. A listing of this routine is presented in Figure A-1(u) of this report. The descriptions of the parameters in the argument list follow.

IO	external file reference number
A	array of numbers to be read
NA	number of elements of A to be read

A machine dependent version of IOREAD is available for CDC machines using input routine BUFFER IN by appropriate replacement of comment cards within the routine.

Called by

BODPAN, VELCMP, SOLVE, SMARCH, FRSTRT

A-25 Subroutine IOWRIT

Subroutine IOWRIT performs an unformatted write to external file, IO. NA consecutive elements of array, A, are written sequentially. This routine is used as a common interface to external files. A listing of this routine is presented in Figure A-1(u) of this report. The descriptions of the parameters in the argument list follow:

IO	external file reference number
A	array of numbers to be written
NA	number of elements of A to be written

A machine dependent version of IOWRIT is available for CDC machines using output routine BUFFER OUT by appropriate replacement of comment cards within the routine.

Called by

BODPAN, VELCMP, SOLVE, NEWRAD, FRSTRT, BODVEL

A-26 Subroutine NETCLC

Subroutine NETCLC calculates the net corner strengths for a given set of wing, pylon, or fuselage constant u-velocity or thickness panels. The net strength for a corner point is calculated by superposing the strengths of the four panels surrounding the point. Whenever a corner point lies on an exterior boundary or along a chordwise row where a break in sweep angle or dihedral angle occurs, surrounding panels are assumed to be present with zero strengths. The superposition scheme is

described in Section 3.3.7 of Reference 3. A listing of the program is presented in Figure A-1(v) of this report.

The panel strengths are passed to the subroutine as formal parameters in the array S, which is dimensioned NCW by MSW. The subroutine first stores this array in a new array P, leaving the first and last columns zero and inserting a zero column whenever a break in leading-edge or trailing-edge sweep angle or dihedral angle occurs. The polar angle acts as dihedral angle for a body interference panel. The first and last rows of P are also set to zero. These zero elements of array P correspond to panels of zero strength mentioned above.

The rest of the subroutine consists of a double DO loop in which the net corner strengths, DP, are calculated according to the following relation:

$$DP(I,J)=P(I,J)-P(I+1,J)-P(I,J+1)+P(I+1,J+1), \quad I=1,\ NCW+1 \quad J=1,\ MSW+k+1 \tag{A-8}$$

where k is the number of breaks in sweep or dihedral which occur over the specified set of panels.

The descriptions of the parameters in the argument list follow:

S	array containing the NCW*MSW panel strengths
DP	array containing the NCW1*MSW1 net corner strengths
NCW	number of panels in a chordwise row; also row dimension of S
NCW1	NCW+1; row dimension of DP
MSW	number of panels in a spanwise row on wing or pylon; number of panels in a ring of fuselage panels
MSW1	MSW+1

MSWK1	number of corner points in a spanwise row of wing or pylon panels; number of corner points in a ring of fuselage panels
PSILE	array containing leading-edge sweep angles for wing or pylon, polar angles for fuselage panels
PSITE	array containing trailing-edge sweep angles for wing or pylon, polar angles for fuselage panels
PHI	array containing dihedral angles for wing, polar angles for fuselage; zero array for pylon

Called by
NULYT

A-27 Subroutine NEWRAD

Subroutine NEWRAD is used to organize the revision of the noncircular body meridian line spacing for the source panels. This routine computes the number of meridional lines to be used for paneling on the half or full body. The allocation of arrays to be used for temporary storage, YB and ZB, are defined also. NEWRAD calls NEWRD2 for the revised meridional spacings for each of the NFUS body segments. If more than one segment exists, the revised ordinates, YB and ZB, are written on external file number 8. A listing of this routine is presented in Figure A-1(v) of this report.

Subroutine references
IOWRIT, NEWRD2

A-28 Subroutine NEWRD2

Subroutine NEWRD2 computes the revised noncircular body meridional line spacing of a given body segment. For each

segment, there are three options for redefining the meridian lines and the option to define the symmetric portion of a body. If KRAD=0, the meridian lines are not changed. If KRAD is positive, the meridian lines are relocated at KRAD equally spaced values of the meridian angle ϕ_k. If KRAD is negative, the user defined values of ϕ_k are used. For each of the above options, new meridian lines may be defined from circular, elliptic, or arbitrary section input. A listing of this routine is presented in Figure A-1(v) of this report.

If the body has circular or elliptic cross section, the y and z coordinates are calculated at each station as follows:

$$
\begin{aligned}
y &= r \cos \phi \\
z &= z_c + r \sin \phi
\end{aligned}
\qquad \text{(A-[illegible])}
$$

where z_c is the camber offset in item 19 of the input and the local radius r is defined as

$$
r = \frac{1}{\sqrt{(\cos\phi/BY)^2 + (\sin\phi/AZ)^2}} \qquad \text{(A-10)}
$$

The parameters BY and AZ are the horizontal and vertical semi-axis of the ellipse. For the circular body, BY and AZ are equal to the radius and the above equation reduces to the circular radius.

If the body has an arbitrary cross section, the y and z coordinates are obtained by linear interpolation at the new values of of the original y and z coordinates read in the program input items 20 and 21.

If the symmetry option is used, IXZSYM=1, the remaining coordinates for the second half of the body are defined. The meridian lines for the symmetric half of the body continue in counterclockwise fashion about the body. The y ordinates are the

negative of the first half of the body, while the z values remain the same.

The descriptions of the parameters in the argument list follow:

YB,ZB	y and z ordinates of the revised meridional spacing
PHIK(K)	ϕ_k, Kth revised meridian for new spacing, deg.
ZFUS(N)	z_C, camber offset of centerline at Nth axial station of segment
FUSBY(N), FUSAZ(N)	BY and AZ, horizontal and vertical semi-axes of ellipse at Nth axial station of segment
SFUS	array containing y and z for arbitrary cross section input option
NFUSOR	number of axial stations of segment
KRAD	number of meridian lines of revised spacing
NRAD	number of meridian lines of original input spacing
IXZSYM	symmetry option

Called by

NEWRAD

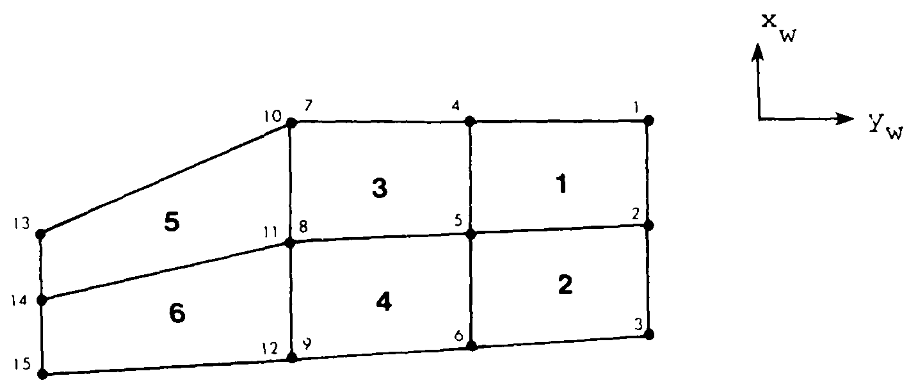
A-29 Subroutine NULYT

Subroutine NULYT selects a subset of constant u-velocity and wing and pylon thickness panel layout data to be saved as an input data set for Program II. The subroutine also calculates and saves the net strengths associated with each panel corner. A listing of the subroutine is presented in Figure A-1(w), and a flow chart in Figure A-8.

The data stored in the new arrays describe a grid of corner points representing the panel layout, thus eliminating some duplication. For example, a corner which is the junction of four

panels is specified four times in the panel coordinate arrays, only once in the corner system. Similarly, the slope of the boundary of two panels adjacent in a chordwise row is saved only once in the new arrays. However, whenever a break in sweep or dihedral angle occurs, an extra chordwise row of points must be saved to accommodate the velocity calculations to be performed in Program II.

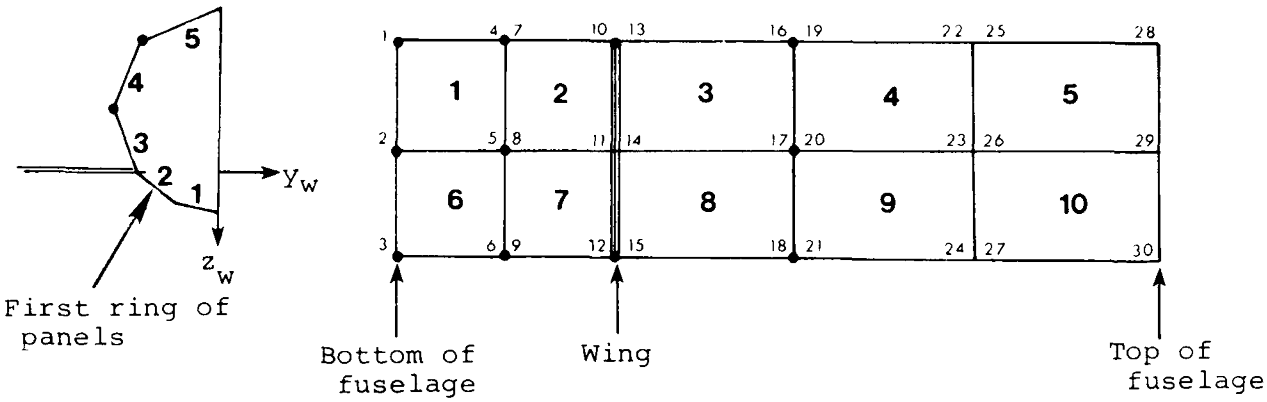
At the beginning of the subroutine indices needed to describe the new arrays are calculated. Next, as the flow chart indicates, data is stored for constant u-velocity panels on the wing. The sketch below illustrates the corner numbering convention as employed for a wing with one leading-edge sweep break and six panels. Panel numbers are in the center of the panels.



For each chordwise row of points, the variables YPT,ZPT,SPHI,CPHI are stored only once. Variables XPT and SWP are stored for all points. The slope and dihedral values used correspond to panels to the left of the points, except for an extra row of points which occurs at a break in sweep or dihedral. For this type of chordwise row (points 7,8,9 in the above sketch), SPHI, CHPI, and SWP correspond to panels to the right.

If a pylon is present, the subroutine next stores the corner data for pylon panels. These data are stored in the same arrays

following the wing corner data. Similarly, if a fuselage is present, the corner data for the fuselage panels is saved in arrays following the wing and pylon data. The correspondence between body interference panels and corner point numbering is shown in the sketch below in which NBDCR1=3, NBDCR2=2. One should note that an extra row of points is saved for each chordwise row of panels.



The subroutine next stores the data for wing and pylon source panels, using the same procedure as for wing and pylon constant u-velocity panels. However, the source panel data are stored in separate arrays.

The last section of the subroutine calculates, by means of subroutine NETCLC, the net strength associated with each corner point. Finally at the end of the routine, a factor of π which had been removed from the source panel thickness slopes for boundary condition calculations, is restored.

Subroutine references
NETCLC

Called by

LDCALC

A-30 Subroutine PANEL

Subroutine PANEL's purpose is to calculate direction cosines of the normal vector and the centroid, area, and inclination angles of an arbitrary quadrilateral panel. It is called to compute the geometric properties of each source panel on the noncircular body. A listing of the routine is presented in Figure A-1(y) of this report.

The four corners of the panel are numbered in a clockwise direction. A diagonal vector T_1 connects points 1 and 3, and a diagonal vector T_2 connects points 2 and 4. The normal vector N is obtained by taking the cross product of these diagonal vectors, and the direction cosines determined by calculating the projections of this vector in the body reference coordinate system. The plane of the panel is defined to be perpendicular to the normal vector and to pass through a point whose coordinates are the averages of the coordinates of the four input points. The input points are then projected into the plane of the panel, and transformed to the reference coordinate system. A new panel coordinate system ξ,η is introduced with the average point of the panel as origin. The coordinates of the centroid and the panel area are calculated in this new system, and the centroid transformed to the reference system. Two angles are used to define the panel orientation. The incidence δ is the angle between the x axis and the line of intersection with the panel of a plane passing through the x axis and perpendicular to the panel. The inclination θ is the angle between the y axis and the line of intersection of the panel with the yz plane. These two angles are calculated in terms of the direction cosines of the normal vector.

The descriptions of the parameters in the argument list follow:

J	panel row number
K	panel column number
NP	panel number
XYZ(NP,1), XYZ(NP,6)	array containing vectors of control points, XPT, YPT,ZPT, angles, THET and DELTA, and area for NPth panel
NBODY	total number of panels
XC,YC,ZC	array containing coordinates of corner points of panels
KFUSOR	number of axial stations in segment

Subroutine references

INLTST, INLXYZ

Called by

BODPN2

A-31 Subroutine PANVEL

Subroutine PANVEL organizes the calls to SORPAN for the calculation of the influence of a panel at a field point. This routine computes the transformations and rotations necessary to calculate the influence of an arbitrarily oriented panel at either a field point or at the control point of another arbitrarily oriented panel. A listing of the routine is presented in Figure A-1(z) of this report.

The routine first computes the combined transformations to rotate a panel or its image into the local panel system and back. If the same transformation is to be used for a number of calculations the logical variable, LZERO, is used to skip around this code in subsequent calls. The field point in question is rotated

into the panel coordinate system and a call made to SORPAN to compute the influence coefficient. If the body is symmetric (IXZSYM=0) the above calculations are repeated for the symmetric panel on the opposing side of the X-Z plane. The influence is summed and rotated back into the original body coordinate system.

The descriptions of the parameters in the argument list follow:

UB,VB,WB	component velocity influence coefficients in body coordinate system at field point
AN	resultant normal velocity coefficient of panel at a field point or at another arbitrarily oriented panel
IXZSYM	body X-Z plane symmetry option

Subroutine references
SORPAN

Called by
BODVEL, FLDVL2

A-32 Subroutine PAS001

Subroutine PAS001 performs the L*U decomposition of the positive definite matrix, A. This routine performs no pivot search during decomposition. It does skip unnecessary calculations associated with off-diagonal zeroes. An error return is provided for encountering zero values on the diagonal. The decomposition procedure is equivalent to

$$A = \begin{bmatrix} 1.0 & & & & \text{zero} \\ L_{21} & 1.0 & & & \\ \cdot & & \cdot & & \\ \cdot & & & 1.0 & \\ L_{n1} & \cdot & \cdot & L_{nn-1} & 1.0 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdot & \cdot & u_{1n} \\ & u_{22} & & & \cdot \\ & & \cdot & & \cdot \\ & & & \cdot & \cdot \\ & \text{zero} & & & u_{nn} \end{bmatrix} \quad \text{(A-11)}$$

The decomposed matrix is stored in the location of the original matrix. A listing of the routine is presentea in Figure A-1(aa) of this report.

The descriptions of the parameters in the argument list follow:

A	positive definite matrix with nonzero diagonal elements
N	actual size of matrix, A
NER	diagnostic. If NER greater than zero, decomposition failed because $\lvert A(NER,NER)\rvert < 10^{-20}$
ND	dimensioned size of A

Called by
BODVEL

A-33 Subroutine PAS002

Subroutine PAS002 solves the system of equations [L*U] * X = B by forward and backward substitution. It is assumed that matrix, A, has been decomposed by routine PAS001 or equivalent so that A contains [L*U]. A listing of the routine is presented in Figure A-1(aa) of this report.

The descriptions of the parameters in the argument list follow:

A	coefficient matrix containing [L*U]
B	matrix containing right-hand sides of the linear system [L*U] * X = B. Contains X on output
N	actual size of [L*U] matrix contained in A
NB	number of right-hand side vectors contained in the first NB columns of B
NDA	dimensioned size of A
NDB	dimensioned size of B

Called by
SMARCH

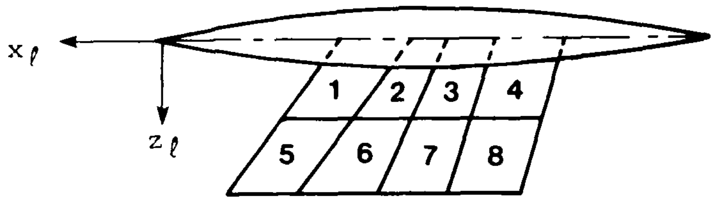
A-34 Subroutine PLYOUT

Subroutine PLYOUT reads in data which describe the geometric characteristics of the pylon and calculates quantities which specify the pylon constant u-velocity panels. The pylon input variables are shown in Figure 8 of Volume II. A listing of the subroutine is presented in Figure A-1(aa) and a flow chart in Figure A-9.

The first part of the subroutine reads in and prints input items 43, 44, and 45. The total number, MP, of pylon constant u-velocity panels is calculated. Next, the subroutine tests whether the pylon leading-edge and trailing-edge sweep angles are constant at all spanwise z-stations. If breaks in sweep occur, the indicator LVSPP is set equal to one; otherwise, LVSPP=0 and the quantity SLPDIF, the difference between leading-edge and trailing-edge slopes for a chordwise row, is computed outside the DO loops. The zero location of the pylon root chord is calculated, and the variables PLEX and CSIDE are initialized for laying out the first chordwise row of panels.

The remainder of the subroutine is a double DO loop within which panel leading-edge and trailing-edge slopes, corner coordinates, and control point coordinates are calculated. These quantities for the pylon are stored in arrays following the same quantities calculated by subroutine WLYOUT for the NPANLS constant u-velocity panels on the wing.

The following sketch shows the numbering convention associated with the pylon panels for a typical pylon-under-wing layout with four chordwise and two spanwise panels.



In the subroutine notation, the panel corners labeled left are those farthest from the pylon root chord.

Called by
LDCALC

A-35 Subroutine PYTHIN

Subroutine PYTHIN reads in the pylon thickness data, input items 46, 47 and 48. A listing of the subroutine is presented in Figure A-1(bb) of this report.

The number of panels in a chordwise row, NCPS, and the number of chordwise rows, MSPS, are first read in along with an index, NUNIP, which indicates whether the thickness distribution is similar at all spanwise stations. The subroutine then reads input item 47. Next, if NUNIP=1, the values of tan θ_p are read in for the first chordwise row and then the values of tan θ_p for the other rows are set equal to those of the first row. If the distribution is not similar, NUNIP=0, the values of tan θ_p for all rows are read in.

Called by
LDCALC

A-36 Subroutine RACKIO

Subroutine RACKIO reads and prints the data which describe and locate the rack and calculates the rack source and doublet distributions. The rack model is limited to description as a circular body. A listing of the routine is presented in Figure A-1(cc) and a flow chart of subroutine RACKIO is presented in Figure A-10 of this report.

The subroutine first reads and prints the dimensional lengths and locations of the rack, and the polynomial coefficients which describe the shape. The rack is then located in the wing coordinate system. Both the line source and doublet distributions for the rack are calculated through a call to BDYGEN. The rack shoulder is located from the first zero slope. Lastly, the rack shock wave shape is computed in a call to SHKSHP.

Subroutine references
    BDYGEN, SHKSHP

Called by
    LDCALC

A-37 Subroutine SHAPE

The purpose of this subroutine is to calculate the body radius and surface slope at a specified axial station. The body shape is specified by a series of polynomials of the form of Equation (1) of Volume II. A flow chart of subroutine SHAPE is presented in Figure A-11 and a listing of the subroutine in Figure A-1(cc).

The calculation performed by this subroutine consists of two steps. The first step is to determine which of the NS polynomials describes the shape at the value of X where the radius and surface

slope are required. Once this is determined, the appropriate set of coefficients is used in Equation (1) of Volume II to determine r/ℓ. The value of dr/dx is found by differentiating Equation (1).

$$\frac{dr}{dx} = \frac{C_7}{2}\left[\frac{2C_2 \frac{x}{\ell} + C_3}{\sqrt{C_2\left(\frac{x}{\ell}\right)^2 + C_3 \frac{x}{\ell} + C_4}}\right] + C_5 + 2C_6 \frac{x}{\ell} \qquad \text{(A-12)}$$

It should be noted that r/ℓ and dr/dx are calculated using the coefficients of the NSth polynomial if x/ℓ is greater than XE(NS).

The quantities in the parameter list are:

X	value of x/ℓ at which radius and surface slope are to be calculated
NS	number of polynomials describing body shape
XE	array containing values of x/ℓ for the end points of the NS polynomials
C	array containing the coefficients of the NS polynomials
R	calculated value of r/ℓ at x/ℓ = X
DRDX	calculated value of dr/dx at x/ℓ = X

Called by
BDYGEN

A-38 Subroutine SHKSHP

Subroutine SHKSHP calculates the nonlinear shape of the shock wave produced by an axisymmetric body at zero angle of attack. The calculations performed here are based on the equations presented in Section 4.1.1 of Reference 3. A listing of this routine is presented in Figure A-1(dd) of this report.

The computation sequence used here is to incrementally compute the change in β with distance from the centerline and numerically integrate the shape radially. The procedure starts at the nose and initializes the slope of the shape to the maximum value BTNOSE prescribed through the input. A new radial distance is prescribed, and an axial search for the maximum radial velocity performed. With that velocity the local Prandtl-Meyer turning angle, local Mach number, and new shock slope, β_{ν_i}, are computed. The new axial location of the shock shape is computed from the average β over the interval step

$$x_{SHK_i} = x_{SHK_{i-1}} + \frac{1}{2} (\beta_{\nu_i} + \beta_{\nu_{i-1}}) \Delta r \qquad \text{(A-13)}$$

The integration procedure is repeated until the calculated β_ν approaches the free stream value as $(1 - \beta_{\nu_i}/\beta_\infty) < 0.01$ or a maximum number of intervals of 50 is reached before terminating.

The descriptions of the parameters in the argument list follow:

NC	maximum number of axial steps used to scan for maximum radial velocity
SX(J)	axial station of Jth line source
SS(J)	line source strength at Jth axial station
BETA	$\sqrt{M_\infty^2 - 1}$, value of free stream
RMAX	maximum body radius; used to determine initial step size, Δr
BTNOSE	limiting value of β at nose; input
NSHK	number of values computed in shock shape table
XSHK	array containing axial location of shock relative to nose
RSHK	array containing radial location of shock relative to centerline

Called by

FUSEIO, RACKIO, STORIO

A-39 Subroutine SMARCH

Subroutine SMARCH solves for the source panel strengths using a ring by ring marching technique. A listing of the subroutine is presented in Figure A-1(dd) of this report.

SMARCH is organized to solve the general source panel solution for a body in supersonic flow:

$$[A]\gamma_B = V_B - [A_I]_B \qquad \text{(A-14)}$$

where [A] is the aerodynamic influence coefficient matrix partitioned into blocks of the coefficients of one ring on another; γ_B are the panel strengths; V_B are the boundary condition normal velocities in the absence of an image body; and $[A_I]$ is the influence coefficient matrix of the image body on the real body control points. No image store effects will be considered in any solutions in the first program. The solution proceeds in blocks of equations, with only those blocks of equations on or below the diagonal computed and saved on TAPE9. The first block corresponding to the influence of a ring on itself is read from TAPE9 in L*U decomposed form. The solution for that ring is computed with PAS002. Subsequent blocks in column form are read and multiplied by the strengths of that ring and subtracted from the boundary conditions for following rings.

The descriptions of the parameters in the argument list follow:

GB	array of panel strengths, γ_B
VB	array of panel boundary conditions destroyed during solution

IA index of starting location in blank common of temporary A matrix

IMAGE logical indicator of presence of image body

Subroutine references

IOREAD, PAS002

Called by

SOLVE

A-40 Subroutine SOLVE

Subroutine SOLVE computes the source panel boundary condition for a noncircular body in the presence of a free stream angle of attack and roll angle and calls SMARCH for the solution for the panel strengths. A listing of the routine is presented in Figure A-1(ee) of this report.

The routine initializes the angle of attack parameters and allocates temporary storage in blank common for the arrays required during the calculation of panel strengths. The boundary conditions are defined as follows

$$V_B = \cos\alpha_c \sin\delta - \cos\delta(\sin\alpha_c \cos\phi_r \cos\theta + \sin\alpha_c \sin\phi_r \sin\theta) \quad (A-15)$$

The solution is then computed in SMARCH. A copy of blank common up through the panel strengths is then saved on TAPE7. SOLVE may be called for multiple angles of attack with each set of strengths saved sequentially after the other.

The description of the parameter in the argument list follows:

IALP index of the number of the IALPth boundary condition to be computed and saved. IALP $\leq$ 7

Subroutine references
    IOWRIT, SMARCH

Called by
    WDYBDY, STORIO

A-41 Subroutine SORPAN

Subroutine SORPAN computes the three components of velocity induced at a specified control point by a constant source distribution on a quadrilateral panel having longitudinal taper and inclined at an angle delta to the free stream direction. This version has been specialized for only supersonic flow. This routine is based on the methods and equations presented in Reference 4. A listing of the subroutine is presented in Figure A-1(ff) of this report.

The descriptions of the parameters in the argument list follow:

UPM,VPM,WPM	three orthogonal components of velocity in local panel coordinates induced by panel with control point XJ,ZJ at field point XI,YI,ZI

Called by
    PANVEL

A-42 Subroutine SOURCE

Subroutine SOURCE calculates coefficients used in the determination of the line source strengths. They occur as terms in Equations (I-14) and (I-17) of Reference 1. The relation of the coefficients to perturbation velocities, $u_{B,a}/V_\infty$ and $v_{B,a}/V_\infty$, induced by a number of line sources distributed along the body centerline is shown in the first two of Equation (I-30) in which

the coefficients occur as the multipliers of K_n. A listing of the subroutine is presented in Figure A-1(gg) of this report. The subroutine is called by subroutine BDYGEN and Section A-4 should be referred to for further details concerning the source strength calculations.

For a specified control point, x_B, r_B, and singularity origin, ξ, the subroutine calculates quantities U and V according to the following equations:

$$U = - \cosh^{-1} \left(\frac{x_B - \xi}{\beta r_B} \right)$$

$$V = \beta \sqrt{\left(\frac{x_B - \xi}{\beta r_B} \right)}$$

At the beginning of the subroutine a test is performed to determine if the control point is ahead of the Mach cone from the source origin. If so, U and V are set tc zero and control is returned to the calling program.

The definition of the variable in the argument list is:

TX ξ; location on body axis of source origin; positive measured from tip of nose

Called by

BDYGEN

A-43 Subroutine STORIO

Subroutine STORIO reads and prints the data which describe and locate the multiple stores and organizes the calculation of the panel or line singularity strengths for both circular and elliptic bodies. The routine may read up to a total number of seven stores of either circular or elliptic shape. Several stores may share the data for the same shape. However, when elliptic stores are

considered, only two elliptic shapes are permitted with up to seven stores of that shape. A listing of the subroutine is presented in Figure A-1(gg) of this report. A flow chart of the routine is presented in Figure A-12.

The subroutine consists of three sequences of DO loops. The first reads and prints NSTRS cards containing the store number, shape, length, maximum radius, location relative to wing, and orientation. The second loop reads and prints NSHPT sets of shape data. If the shape is circular, items 57 through 59 of the input are read for each of the shapes. If the shape is elliptic, items 60 through 75 are read through a call to GEOM. The source panel geometry is computed and a temporary copy of all shape geometry data saved on TAPE11. The location of the store nose in the wing cordinate system is then computed for each of the stores.

The last major loop is a double DO loop over the number of shapes and over the number of stores. The outer loop is over the number of shapes to facilitate reading the panel geometry saved on TAPE11. In the inner loop on the number of stores, the store shape number is compared with the shape under consideration and the appropriate properties computed. For circular shapes, BDYGEN is called to compute the line source and doublet distributions, the shoulder location and shock shape are then calculated. For elliptic shapes, VELCMP is called to generate the influence coefficient matrix. Two calls to SOLVE may be made. For the first store of the shape the panel strengths are computed at zero degrees angle of attack and the nonlinear shock shape computed in BSHOCK. For each of the stores of that shape SOLVE is called again to compute the panel strengths at the angle of attack and roll of the fixed store. A copy of the store indices, geometry arrays, and strengths are then saved on TAPE10 by FRSTRT.

Subroutine references

BDYGEN, BSHOCK, FRSTRT, GEOM, SHKSHP, SOLVE, VELCMP

Called by

LDCALC

A-44 Subroutine SWNGIN

Subroutine SWNGIN reads in data required to describe the geometric characteristics of the wing and to lay out the wing constant u-velocity panels. A listing of the subroutine is presented in Figure A-1(ii) of this report.

The first part of the subroutine reads in and prints input items 35, 36, and 37, which consist of wing geometry data and quantities used to locate the trapezoidal-shaped elemental panels. If any nonzero dihedral angle is input, the logical variable ZDIHED is set equal to FALSE. Next, the wing twist and camber distribution, if any, is read, input items 38 and 39. Two indices, NTAC and NUNI, are first input. If NTAC=0 there is no twist and camber. The index NUNI indicates whether the twist and camber distribution is similar at all spanwise stations. If it is similar, NUNI=1, the values of tan α_ℓ are read in for the first row and then the values of tan α_ℓ for the other rows are set equal to those of the first row. If the distribution is not similar, NUNI=0, the values of tan α_ℓ for all rows are read in.

Called by

LDCALC

A-45 Subroutine THKLYT

Subroutine THKLYT calculates quantities which characterize the wing and pylon thickness source panels. These quantities are

stored in arrays in each of which the wing panel variables precede the pylon variables. All panel coordinates are expressed in the wing coordinate system which is shown in Figure 7 in Volume II. A listing of the subroutine is presented in Figure A-1(ii) and a flow chart in Figure A-13.

The first part of the subroutine calculates the layout of the wing panels. After the variables CSIDEP, SLPDIF, ZLER, and WLEX are initialized for the first chordwise row of panels, the remaining calculations are performed within a double DO loop. The outer loop index, I, specifies the chordwise row and the inner loop index, K, the panel location in the Ith row. Within the inner loop the panel leading-edge and trailing-edge slopes, the sine and cosine of the dihedral angle, and the corner coordinates are calculated and stored. The panels are numbered consecutively in chordwise rows beginning with panel number one of row one adjacent to the wing root chord at the leading edge. The sequence proceeds in increasing numbers to the trailing edge, then back to the leading edge for the second chordwise row. The process continues until the last panel, numbered MS, is located adjacent to the wing tip at the trailing edge.

If a pylon is present (NPY=1), the remainder of the subroutine calculates the panel leading-edge and trailing-edge slopes and the corner coordinates for the pylon source panels. The procedure used differs very little from the wing panel calculations. Values of YPL, ZPL, and LVSPP, previously calculated in subroutine PLYOUT, are used. Also, the initial value of PLEX depends upon the pylon location; the index IP is tested for this purpose. The numbering convention associated with the pylon thickness panels is the same as that used for pylon constant u-velocity panels. It is described in Section A-34. In describing corner coordinates for both wing and pylon panels, the corners closer to the respective root-chords

are designated right corners; those farther from the root-chords are designated left corners.

Called by

LDCALC

A-46 Subroutine THKOUT

Subroutine THKOUT prints the slopes of the wing and pylon thickness distributions which were read in as items 42 and 48 of the input data, see Section 3.2.2. After the thickness slopes are printed, they are divided by π and saved in a combined array, DZDX, for subsequent velocity calculations. A listing is presented in Figure A-1(jj) of this report.

Called by

LDCALC

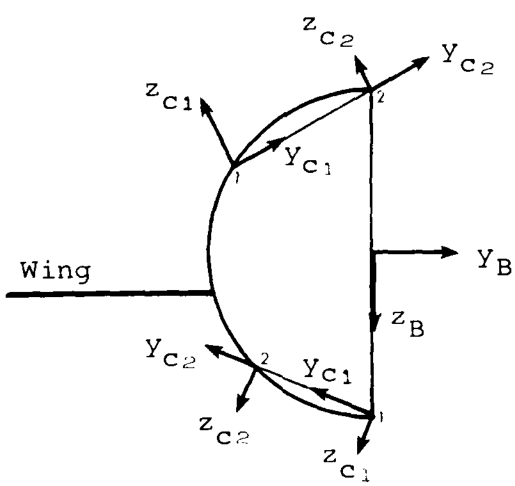
A-47 Subroutine VELBD1

Subroutine VELBD1 is used to obtain the single-panel u-velocity panel influence coefficients which occur indirectly in Equations (24), (25), and (26) of Reference 3. Section A-13 of this report should be consulted for further details concerning these coefficients. A listing of the subroutine is presented in Figure A-1(kk), and a flow chart in Figure A-14 of this report. The subroutine uses quantities calculated in subroutine BLYOUT which is described in Section A-5. The wing coordinate system is shown in Figure 7 in Volume II.

At the beginning of the subroutine the logical variable PYPNL is set equal to FALSE, the velocity totals UP, VP, and WP are initialized to zero, and the panel leading-edge and trailing-edge

slopes, EM, are defined as zero. In the remainder of the subroutine the influence functions for the four corners of the Ith panel are calculated and superposed. The superposition scheme is described in Section 3.3.4 of Reference 3 and that reference should be consulted for further details. In the corner numbering convention for fuselage panels corner one is the left front, corner two the right front, corner three the left rear, and corner four the right rear panel corner. The right corners are clockwise from the left when viewed from the rear.

Next in the subroutine the corner influence function totals TU, TV, and TW are initialized to zero and a test is performed to determine if the point at which velocities are to be calculated lies ahead of the panel leading edge. If so, all calculations for the panel are skipped. If the point is not ahead of the leading edge, a test is performed next to determine in which fuselage quadrant the panel lies. As the flow chart indicates, two similar but distinct transformation and superposition procedures are followed, depending upon the panel location. In each procedure, the field point is first located relative to corner one in a corner coordinate system which is shown in the sketch below. Each panel corner has an associated coordinate system. The x_c axis, not shown in the sketch, is positive to the rear.



If corner one is in the upper left quadrant the sign of the y_c coordinate of the field point is reversed and subroutine VELO1 is called to calculate the influence of the corner on the image of the point with respect to the x_c, z_c plane. The sign of V, which is the returned influence function in the y_c direction, is then reversed. The functions U,V,W are resolved back into the wing system and superposed in the same manner as that used for wing panels with positive sweep (see Figure 4, Reference 1). Panels in the lower left quadrant are treated in the same manner as wing panels with negative sweep. No further coordinate change is necessary and subroutine VELO1 returns U,V, and W which are resolved into the wing system and superposed.

Next, in each procedure, the influence of the mirror image of corner one with respect to the aircraft vertical plane of symmetry is calculated. This is accomplished by the equivalent method of calculating the direct influence of corner one on the field point image. Following the call to subroutine VELO1 the sign of V is reversed in the superposition.

Corner one calculations are repeated in a similar manner for corners two, three, and four, but only corner one is detailed in the flow chart. After superposition of the four corner influence functions for the panel is completed, the influence coefficients for the fuselage panel at the given control point are returned by the subroutine.

Subroutine references

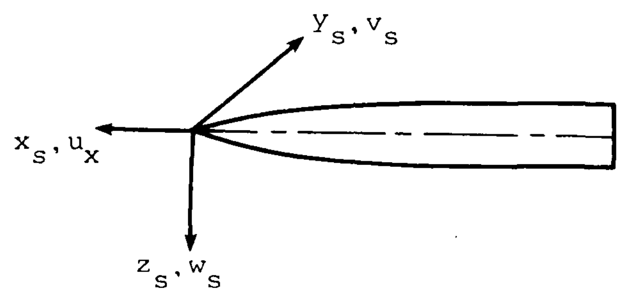
VELO1

Called by

DPCOEF

A-48 Subroutine VELCAL

Subroutine VELCAL calculates perturbation velocities at a given field point due to the fuselage and store source and doublet distributions. A listing of the subroutine is presented in Figure A-1(mm) of this report. The fuselage coordinate system is shown in Figure 5 in Volume II. The store coordinate system is shown in the following sketch.



The coordinates of the field point are given as formal parameters in the appropriate body coordinate system. The subroutine first transforms these coordinates into the VELCAL system by changing the sign of X and Z, and then into the polar coordinates XFIELD, RFIELD, and THETA.

The major part of the program consists of a DO loop within which the axial, radial, and tangential velocities due to the N sources and doublets are calculated and summed. A test is made to determine whether the field point is ahead of the Mach cone from the Ith source origin, TX(I). If so, all remaining perturbation velocities are equal to zero and no further calculations are performed within the loop. At the end of the subroutine, these velocities in the VELCAL system are resolved back into the body coordinate system.

The variables in the subroutine argument list are:

T	array containing the source strengths
TC	array containing the doublet strengths
TX	array containing the x locations of the origins of the singularities; positive, measured aft from tip of nose
N	number of line sources and doublets
XP	x coordinate of field point in body system
Y	y coordinate of field point in body system
ZP	z coordinate of field point in body system
U1	u/V_∞ perturbation velocity at field point; body system
V1	v/V_∞ perturbation velocity at field point; body system
W1	w/V_∞ perturbation velocity at field point; body system
BODYL	body x-station at which base singularities originate, feet
SUMK	strength of source originating at body base
SUMKD	strength of doublet originating at body base

Called by

DPRHS

A-49 Subroutine VELCMP

Subroutine VELCMP organizes and calls for the calculation of the aerodynamic influence coefficient matrices for the body source panels. This routine optionally reads the Mach number, angle of attack, and angle of roll. It initializes all Mach number parameters, rewinds any external files, and allocates temporary storage locations in blank common to hold blocks of influence coefficients. VELCMP performs the loop over the number of body segments calling BODVEL to compute the influence coefficients for the panels within

a segment on all other control points. A listing of this subroutine is presented in Figure A-1(nn) of this report.

The parameter in the argument list is:

IREAD	option index whether to read the Mach number and angle of attack IREAD=0, read XMACH,ALPHAC,PHIR IREAD=1, no input read; parameters must be in common

Subroutine references
BODVEL, IOREAD, IOWRIT

Called by
WDYBDY, STORIO

A-50 Subroutine VEL01

Subroutine VEL01 calculates the aerodynamic influence functions of a semi-infinite triangle associated with a constant u-velocity panel, as described in Section II-2.1 of Appendix II of Reference 1. The influence functions relate the panel singularity strength to the perturbation velocities induced by the triangle at a given point. They occur as the coefficients of $1/\pi(u_+/V_\infty)$ in Equations (II-4) and (II-12) of Reference 1. A listing of the subroutine is presented in Figure A-1(nn), and a flow chart in Figure A-15 of this report. The coordinate system used by the subroutine is shown in Figure 3 of Reference 1.

At the beginning of the subroutine the logical variable FELT is initialized to TRUE and a test is performed to determine if the field point is located ahead of the influencing triangle ($X \leq 0$).

If so, the influence functions U,V,W are set to zero, FELT is set to FALSE, and control is returned to the calling program.

Next, the variable PYPNL is tested and, if the triangle is on the pylon, a transformation is performed which rotates the triangle into the VELO1 x,y plane. After the calculation of the logical variable INSIDE and other frequently used quantities, the remainder of the subroutine consists of four major sections in which the influence function terms, F1, F2, F4, F5, and F7, are calculated. Each section corresponds to a condition of the slope, EML, of the leading edge associated with the semi-infinite triangle. The subroutine requires that EML $\geq$ 0 and this is accounted for in the VELO1 calling programs. The four leading-edge conditions are described fully, with accompanying sketches in Section II-2.1 of Reference 1. All equation numbers mentioned in the following paragraphs are from Section II-2.1 of Reference 1.

The first section of the subroutine corresponds to a subsonic leading edge, BTSQ < EMLSQ. Equation (II-5) is used if the point is inside the Mach cone from the origin, INSIDE=TRUE. If not, U,V, and W are set to zero. In this section as in the remaining ones, discontinuities in some of the equations may occur for certain field point locations. In such cases the affected influence function is set to zero. The quantities YYEDGE and TLRNC, as well as Y and Z, are used to test the singularity locations.

If BTSQ=EMLSQ, the leading edge is a sonic leading edge. The equations used are the same as for the subsonic case except for the function F2, which is given by Equation (II-7). If the point lies outside the Mach cone from the origin, the influence functions equal zero.

The third section of the subroutine is used if the triangle leading edge is supersonic, BTSQ > EMLSQ. Equations (II-5) and (II-9) calculate the terms of the influence functions if INSIDE= TRUE. If not, a second test is performed and Equation (II-11) is used if the point is inside the Mach cone whose origin is on the leading edge at the field point y location, otherwise the functions U,V,W are set to zero.

The fourth section of the subroutine is executed if the leading edge is unswept, EML=0. For this special case the perturbation velocity equations are given by (II-12). If INSIDE=TRUE, the influence function terms are given by (II-5) and (II-13). Outside the Mach cone from the origin but inside the cone from the leading edge Equation (II-14) is used, otherwise, U,V,W are set to zero.

The last part of the subroutine calculates the functions U,V, W from the component terms, in the case of a leading edge with positive sweep, using Equation (II-4). If the triangle is located on the pylon, V and W are rotated back into the pylon u-velocity panel coordinate system.

Called by

VELBD1, VELPP1, VELWP1

A-51 Subroutine VELOT1

Subroutine VELOT1 calculates the aerodynamic influence functions of a semi-infinite triangle associated with a wing or pylon thickness panel, as described in Section II-2.2 of Appendix II of Reference 1. The influence functions relate the panel source strength to the perturbation velocities induced by the triangle at a given point. They occur as the coefficients of $1/\pi(\tan \theta)$ in Equations (II-15) and (II-16) of Reference 1. A listing of the subroutine is presented in Figure A-1(pp) of this report. The

subroutine is very similar in logic to subroutine VELO1 which is described in detail in Section A-50 and represented by a flow chart in Figure A-15.

In subroutine VELOT1, three component terms, F1, F2, and F5, need to be calculated in order to determine the influence functions UTH, VTH, and WTH. Referring to Sections II-2.1 and II-2.2 of Reference 1, function F1, F2, and F5 are specified in Equation (II-5) for the case of a subsonic leading edge, in Equations (II-5) and (II-7) for a sonic leading edge, and in Equations (II-5), (II-9), and (II-11) for a supersonic leading edge. For the special case of an unswept leading edge (EML=0), the general perturbation velocity equations are given by (II-16). If the given point lies inside the Mach cone from the origin of the triangle, function F1 is given by Equation (II-13), function F5 by Equation (II-5), and function F2 by Equation (II-17). If the point lies outside the Mach cone from the origin but inside the Mach cone from the triangle leading edge at the field point y location, functions F1, F2, and F5 are given by Equation (II-16). In all leading edge cases, the function F1, F2, and F5 are singular for certain field point locations. When this occurs, the affected influence function is set to zero. All influence functions are set equal to zero if the point lies in the plane of the semi-infinite triangle (ZTH=0).

A table equating the algebraic and program notation for variables in common for subroutine VELOT1 is presented in Appendix B-2 of this report. Almost all of the notation used in subroutine VELOT1 is defined in this table. One should note that the influence functions, U,V,W and the point coordinates, YS, ZS, in VELO1 are named UTH, VTH, WTH, YTH, and ZTH, respectively, in subroutine VELOT1.

Called by

VELOT1, VELPT1

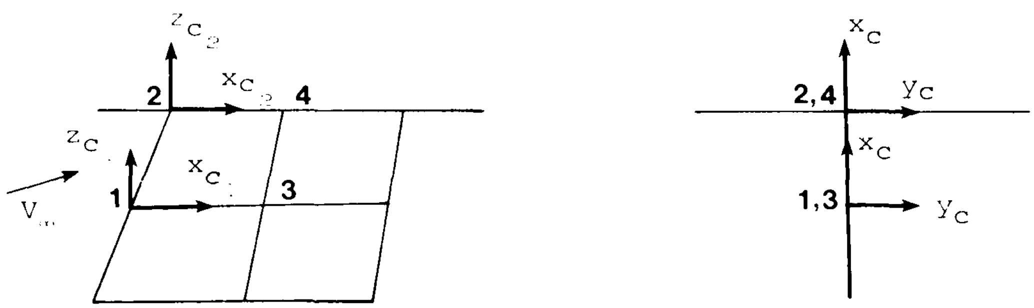
A-5. Subroutine VELPP1

Subroutine VELPP1 calculates the single panel influence coefficients which form the coefficients of $1/\pi(u_+/v_\infty)$ in Equation (12) of Reference 1. A listing of the subroutine is presented in Figure A-1(qq) and a flow chart in Figure A-16. The quantities XX, YY, ZZ in the subroutine argument list are the x,y,z coordinates in the wing coordinate system. The wing coordinate system is shown in Figure 7 of Volume II.

At the beginning of the subroutine, the logical variable PYPNL is set equal to TRUE, indicating to subroutine VELO1 that calculations are to be performed for a pylon panel. The quantities YDIR and YIMG are calculated and the velocity totals, UP, VP, WP, are initialized to zero. In the remainder of the subroutine the influence functions for the four corners of the panel are calculated and superposed. The superposition scheme is described in Section 3.3.4 of Reference 3 and that reference should be consulted for further details. The corner numbering convention for pylon panels associates corners one and two with the leading edge left and right corners, respectively; corners three and four with the trailing edge left and right corners, respectively. The left corners are those farther from the pylon root chord.

At the beginning of the calculation the influence function totals are initialized to zero and the leading-edge slope, EM1, and trailing-edge slope, EM2, are defined. Next, a test of the sign of EM1 is performed and two distinct transformation and superposition procedures are followed depending on the results of this test, as the flow chart indicates. In each procedure a test is

performed first to determine if the field point lies ahead of the most forward leading-edge corner; if so, all calculations for the Ith panel are skipped. If the point is not ahead of the leading edge, the point is located relative to corner one in a corner coordinate system which is illustrated in the sketch below. Each panel corner has an associated coordinate system.



For a panel with swept back leading edge, EM1 $\geq$ 0 the superposition scheme is the same as for a wing panel with positive sweep (see Figure 4, Reference 1). The field point z coordinate is reversed and subroutine VELO1 is called to calculate the influence of corner one on the image of the point with respect to the x ,y plane. The sign of W is then reversed. The functions U,V,W are resolved back into the wing system and superposed. If the panel leading edge is swept forward, superposition is the same as for wing panels with negative sweep. The sign of EM1 is reversed and subroutine VELO1 returns U,V,W which are resolved into the wing system and superposed. It should be noted that the subroutine code combines superposition and transformation steps and that the final sign changes of U and W are made at the end.

The next calculations are omitted if the pylon is located under the fuselage centerline. Otherwise, the influence of the mirror image of corner one with respect to the vertical plane of symmetry

is calculated. This is accomplished by the equivalent method of calculating the direct influence of corner one on the image of the field point and then reversing the sign of V.

Corner one calculations are repeated in a similar manner for corners two, three, and four, but only corner one is described in detail in the flow chart. After superposition of the four corner influence functions for the panel is completed the influence coefficients for the Ith pylon panel and the given control point are returned by the subroutine.

Subroutine references

VELO1

Called by

DPCOEF

A-53 Subroutine VELPT1

Subroutine VELPT1 calculates perturbation velocities at a field point due to the pylon thickness distribution according to methods described in Section 3.3 of Reference [illegible]. The program logic is very similar to that of subroutine VELPP1 which calculates velocities induced by the pylon constant u-velocity panels. Subroutine VELPP1 has been described in detail in Section A-52 and is represented by a flow chart in Figure A-16 of this report. Only those details, therefore, in which subroutine VELPT1 differs from subroutine VELPP1 are included in this description. A listing of the subroutine is presented in Figure A-1(ss). The quantities in the subroutine argument list are the x,y,z coordinates of the field point in the wing coordinate system.

The coordinates of the field point are given as formal parameters in the wing coordinate system. The point is located relative to each of the four panel corners using the same transformation scheme as in VELPP1. However, the corner coordinate arrays which define the pylon thickness panels and which have been previously calculated by subroutine THKLYT (see Section A-45) are used in the transformations. Subroutine VELOT1 is called to calculate the corner influence functions, U,V,W, which are then superposed in the same manner as in VELPP1.

The subroutine calculates perturbation velocities only. The influence coefficients are multiplied by DZDX(I) to obtain perturbation velocities induced by the Ith panel at the given point. These velocities are calculated and summed for all pylon thickness panels.

Subroutine references
VELOT1

Called by
DPRHS

A-54 Subroutine VELWP1

Subroutine VELWP1 calculates the influence coefficients at a specified control point due to a wing constant u-velocity panel. The influence coefficients occur in Equation (27) of Reference 3 as the coefficients of $/\pi)(u_+/V_\infty)$. A listing of the subroutine is presented in Figure A-1(tt), and a flow chart in Figure A-17. The quantities in the subroutine argument list are the x,y,z coordinates of the field point in the wing coordinate system. The wing coordinate system is shown in Figure 7 of Volume II.

At the beginning of the subroutine, the logical variable PYPNL is set equal to FALSE indicating to subroutine VEL01 that calculations are for a wing rather than a pylon panel. The influence function totals, TU, TV, TW, are initialized to zero. The leading edge slope, EM1, and the trailing edge slope, EM2, as well as the dihedral angle sine and cosine, are defined. The remainder of the subroutine consists of the calculation and superposition of the influence functions for the four corners of the influencing panel. The superposition scheme is described in Section 3.3.4 of Reference 1 and that reference should be consulted for further details. The corner numbering convention for wing panels is shown in Figure 4 of Reference 1.

Two distinct transformation and superposition procedures are followed depending on the sign of EM1, for corners one and two, and EM2, for corners three and four. In each procedure the point is first located relative to corner one in the coordinate system associated with subroutine VEL01, illustrated in Figures 3 and 4 of Reference 1. Dihedral effects, if nonzero, are included in the coordinate transformations. Subroutine VEL01 is called to calculate the corner influence functions, U,V,W, which are resolved back into the wing system and superposed. Superposition and transformation steps are combined in the code and the final sign changes of U and W are made at the end of the subroutine. Next, in each procedure, the influence of the image of corner one with respect to the aircraft vertical plane of symmetry is calculated. This is accomplished if EM1 < 0 by the equivalent method of calculating the direct influence of corner one on the field point image and then reversing the sign of V.

Corner one calculations are repeated in a similar manner for corners two, three, and four, but only corner one is detailed in the flow chart.

Subroutine references
VELO1

Called by
DPCOEF

A-55 Subroutine VELWT1

Subroutine VELWT1 calculates perturbation velocities at a given field point due to the wing thickness distribution, according to methods described in Section 3.3 of Reference 3. The program logic is very similar to that of subroutine VELWP1, which calculates influence functions for the wing constant u-velocity panels. Subroutine VELWP1 has been described in detail in Section A-54 and is represented by a flow chart in Figure A-17 of this report. Only those details, therefore, in which subroutine VELWT1 differs from subroutine VELWP1 are included in this description. A listing of the subroutine is presented in Figure A-1(ww). The quantities in the subroutine argument list are the x,y,z coordinates of the field point in the wing coordinate system.

The coordinates of the field point are given as formal parameters in the wing coordinate system. The point is located relative to each of the four corners of the trapezoidal shaped thickness panels using the same transformation schemes as in VELWP1. However, the corner coordinate arrays which define the wing thickness panels and which have been previously calculated by subroutine THKLYT (see Section A-45) are used in the transformations. Subroutine VELOT1 is called to calculate the corner influence functions, U,V,W, which are then superposed in the same manner as in VELWP1.

Finally, the subroutine calculates perturbation velocities rather than single panel influence coefficients. Thus, after

the calculation of the influence coefficients for the Ith panel is completed, the influence coefficients are multiplied by DZDX(I) to obtain the perturbation velocities induced by the Ith panel at the given field point. These velocities are calculated and summed for all wing thickness panels.

Subroutine references
VELOT1

Called by
DPRHS

A-56 Subroutine WDYBDY

Subroutine WDYBDY reads and prints the noncircular fuselage data and organizes the calling sequence for the body panel layout, the computation of the source strengths, and the generation of the nonlinear shock shape. A listing of the routine is presented in Figure A-1(zz) of this report.

WDYBDY is used to generate the required solutions for the noncircular fuselage. It first reads and prints the basic length, width, nose shock angle, and interference shell length and number of panels. The noncircular fuselage geometry is read and panels laid out in a call to GEOM. The panel cross section data to be used for the fuselage interference shell is saved by YZBIP. The influence coefficient matrices and the solution for the shock strength are generated in calls to VELCMP and SOLVE. The nonlinear shock shape is generated in a call to BSHOCK and all data to be passed to Program II saved on TAPE10 by FRSTRT.

Subroutine references
BSHOCK, FRSTRT, GEOM, INLSHK, SOLVE, VELCMP, YZBIP

Called by

LDCALC

A-57 Subroutine WITHIN

Subroutine WITHIN reads in the wing thickness data, input items 40, 41 and 42. A listing of the subroutine is presented in Figure A-1(aaa) of this report.

The number of panels in a chordwise row, NCWS, and the number of chordwise rows, MSWS, are first read in along with an index, NUNIS, which indicates whether the thickness distribution is similar at all spanwise stations. The subroutine then reads input item 41. Next, if NUNIS=1, the values of tan θ are read in for the first chordwise row of panels and then the values of tan θ for the other rows are set equal to those of the first row. If the thickness distribution is not similar, NUNIP=0, the values of tan θ for all rows are read in.

After the thickness slopes are read in, an input error check is performed. If any of the tan θ values is zero or negative at the wing leading edge, an error message is printed (see Section 3.5, Volume II) and the program halts.

Called by

LCALC

A-58 Subroutine WLYOUT

Subroutine WLYOUT calculates quantities which characterize the constant u-velocity panels on the wing. An example of a wing trapezoidal panel is shown in Figure 2 of Reference 1. Similar routines PLYOUT and BLYOUT calculate these quantities for the pylon and fuselage panels, respectively. The arrangement

of variables in any single coordinate array is wing panels first, pylon panels second, and fuselage panels last. All coordinates are in the wing coordinate system which is shown in Figure 7 of Volume II. A listing of the subroutine is presented in Figure A-1(aaa), and a flow chart in Figure A-18 of this report.

As indicated by the flow chart, the first part of the subroutine tests whether the wing leading-edge and trailing-edge sweep angles are constant at all spanwise Y-stations. If breaks in sweep occur, the indicator LVSWP is set equal to one; otherwise, LVSWP=0 and the quantity SLPDIF, the difference between leading-edge and trailing-edge slopes for a chordwise row, is computed outside the main DO loop.

After initializing the quantities CSIDEP, ZLER, and WLEX, the remainder of the subroutine consists of a double DO loop. The outer loop index, I, controls the chordwise row; the inner loop index, K, specifies the panel location in the Ith row. Within the inner loop, the leading-edge and trailing-edge slopes, the sine and cosine of the dihedral angle, the corner coordinates, and the control point coordinates are calculated and stored for each panel. The convention used in numbering the NPANLS wing constant u-velocity panels is the same as that used for the wing thickness panels. Section A-45 should be referred to for details. In the subroutine notation, the right-hand side of the panel is the one closer to the root-chord.

Called by

LDCALC

A-59 Subroutine WRFILE

Subroutine WRFILE writes the data set on TAPE12 which contains all the information required to continue the supersonic store

separation computations in Program II. A listing of the routine is presented in Figure A-1(bbb) of this report.

WRFILE writes all the data required for Program II unformatted on TAPE12. The data includes the list of headings, control indices, various constants, u-velocity data, thickness panel data, fuselage geometry and strength data, rack data, pylon data, and data for multiple stores. When saving information required for the noncircular fuselage or elliptic stores, the appropriate arrays are transferred from TAPE10 to TAPE12 using routine FRSTRT.

Subroutine references

FRSTRT

Called by

LDCALC

A-60 Subroutine YZBIP

Subroutine YZBIP is used to organize the scan of the source panel geometry to define the cross section data for the non-circular fuselage interference shell. YZBIP performs the loop over the number of body segments in organizing the data for a scan of the axial geometry. The scan of panel geometries within a segment are performed in a call to YZBIP2. A listing of this subroutine is presented in Figure A-1(ccc) of this report.

Subroutine references

YZBIP

Called by

WDYBDY

A-61 Subroutine YZBIP2

Subroutine YZBIP2 scans the source panel geometry of a single body segment and performs the interpolation in the data to define the y-z values of the section to be used by BLYOT2 to define the body interference shell. The routine scans the arrays of axial stations to find the two sections which bound the input station, XBIP (item 15). A linear interpolation between the Y-Z values of the two sections is then performed. This table of Y-Z values is used by BYLOT2 to specify the Y-Z values of the fuselage interference shell as close to the source panel geometry as possible. A listing of this routine is presented in Figure A-1(ccc) of this report.

The descriptions of the parameters in the argument list follow:

XB	array of axial stations of section data of segment
YB,ZB	arrays of Y-Z coordinates of panel corners
KRAD	number of corners in the polar direction about the body
NFUSOR	number of axial stations in segment

Called by

YZBIP

Figure A-1.- Listing of Program I

(Pages 84 through 138)

```
      PROGRAM LDCALC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE12
     * ,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11)
C
C     PROGRAM TO CALCULATE SIX-DEGREE-OF-FREEDOM STORE TRAJECTORIES
C           AT SUPERSONIC SPEEDS
C
C     THIS IS PROGRAM 1
C
C     THIS PROGRAM LAYS OUT THE PANELS REPRESENTING THE PARENT AIRCRAFT,
C         CALCULATES BODY SOURCE AND DOUBLET STRENGTHS, AND THE CONSTANT
C         U VELOCITY PANEL STRENGTHS.
C
C     ALL QUANTITIES READ IN BY THIS PROGRAM OR CALCULATED BY THIS
C         PROGRAM WHICH ARE REQUIRED BY THE TRAJECTORY PROGRAM,
C         PROGRAM 2, ARE WRITTEN OUT ON TAPE 12.
C
C     THE TRAJECTORY CALCULATION PROGRAM READS THIS DATA SET IN.
C
      LOGICAL CENTER
C
C     DIMENSION STATEMENTS
C
      DIMENSION HEAD(20),FVN(200,20))
C
C     COMMON STATEMENTS
C
      COMMON /BPGEOM/ SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1     SANGW
      COMMON /CAMBER/ ALPHAL(200)
      COMMON /CONFIG/ NFU,NPY,NSTRS,LVSWP,NRACK
      COMMON /CONSTS/ PI,PI2,DTOR,RTOD,FOURPI
      COMMON /EXVEL / UEI(200),VEI(200),WEI(200),CIR(200)
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETAO,BETASQ,FMCHSQ
      COMMON /FSGEOM/ FRMAX,SSPAN,   FLTHC,BODYPL
      COMMON /ICVEL / UP,VP,WP,II,IF,        DELTP(200)
      COMMON /INDEX / NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP
     1   ,NCP,MSP,NIP,N2,N2P,NPTOT
      COMMON /PYGEOM/ Z(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
     1     SLLE,PSLPDF,CENTER,ZPL,LVSPP
      COMMON /RKGEOM/ RRMAX,RLTHC,XWROC,YWRO,ZWRO,NRPOLY,RXEND(7),
     1       RCOEF(7,7),XBRO,YBRO,ZBRO,RIBCR,SRIBCR,CRIBCR
      COMMON /SHPDAT/ NFPOLY,FXEND(7),FCOEF(7,7)
      COMMON /STGEOM/ SRMAX(7),SLTHC(7),XWSOC(7),YWSO(7),ZWSO(7),
     1      NSPOLY(7),SXEND(7,7),SCOEF(7,7,7),XBSO(7),YBSO(7),ZBSO(7),
     2      SIBCR(7),NUMSTR(7),SSIBCR(7),CSIBCR(7),SPHIRR(7),NSHAPE(7)
     3      ,NSHPT,MSHAPE(7)
      COMMON /THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400)
     1   ,XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2   THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3   YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4   PSPSTE(20),SNPHS(400),CSPHS(400)
      COMMON /VELARG/ X,DUMY,DUMZ,U,V,W,EM,TLRNC,TIPY,PYPNL
      COMMON /WGEOM / XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1     PSIWTE(20),Y(20),PHID(20),ZDIMED,WICR
      COMMON /WPGEOM/ XRF(200),XRB(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2     SNPHI(200),CSPHI(200)
      COMMON            A(40200)
      EQUIVALENCE (FVN(1,1),A(1))
C
C     FORMAT STATEMENTS -------------------------------------------------
C
  701 FORMAT(10I5)
  702 FORMAT(1H1,36X,51HSUPERSONIC SIX-DEGREE-OF-FREEDOM TRAJECTORY PROG
     1RAM//)
  703 FORMAT(20A4)
  704 FORMAT(10X,20A4)
  705 FORMAT(/1H )
  706 FORMAT(8F10.0)
  707 FORMAT(10X,26HAIRCRAFT FLIGHT CONDITIONS/15X,17HANGLE OF ATTACK =,
     1F6.2,8H DEGREES/15X,13HMACH NUMBER =,F5.2//)
  712 FORMAT(1H1,10X,15HWING INPUT DATA)
  713 FORMAT(/15X,66HLOCATION OF WING ROOT CHORD LEADING EDGE RELATIVE T
     10 FUSELAGE NOSE/25X,4HXF =,F10.5,5H FEET/25X,4HZF =,F10.5,5H FEET
     2/15X,60HINCIDENCE ANGLE OF WING ROOT CHORD RELATIVE TO FUSELAGE AX
     315/25X,4HIW =,F8.2,8H DEGREES)
  740 FORMAT(1H1,45X, 43HU-VELOCITY PANEL CONTROL POINT COORDINATES,/
     1 47X,46HINTERFERENCE VELOCITIES INDUCED AT THESE POINTS,/
     2 45X,45HINPUT TWIST AND CAMBER SLOPE AT THESE POINTS,/
     3 52X,31HAND PANEL SINGULARITY STRENGTHS/
     4 55X,24H(WING COORDINATE SYSTEM))
  741 FORMAT(/10X,19HWING CONTROL POINTS/17X,3HROW,2X,5HPANEL,5X,5HX, FT
     1,6X,5HY, FT,6X,5HZ, FT,6X,6HU/VINF,5X,6HV/VINF,5X,6HW/VINF,5X,
     26HALPHAL,3X,10HUPLUS/VINF)
  742 FORMAT(15X,2I5,3X,8F11.5)
  743 FORMAT(/10X,20HPYLON CONTROL POINTS/17X,3HROW,2X,5HPANEL,5X,5HX, F
     1T,6X,5HY, FT,6X,5HZ, FT,17X,6HV/VINF,25X,10HUPLUS/VINF)
  744 FORMAT(15X,2I5,3X,3F11.5,11X,F11.5,22X,F11.5)
  745 FORMAT(/10X,23HFUSELAGE CONTROL POINTS/16X,4HRING,2X,5HPANEL,5X,
     15HX, FT,6X,5HY, FT,6X,5HZ, FT,17X,6HV/VINF,5X,6HW/VINF,14X,
     210HUPLUS/VINF)
  746 FORMAT(15X,2I5,3X,3F11.5,11X,2F11.5,11X,F11.5)
  754 FORMAT(///5X,20H***** NSTRS INPUT AS,I3,33H, PROGRAM LIMITED TO SE
     1VEN STORES)
C
      PI=3.141592653
      PI2=1.570796327
      DTOR=0.0174532925
      RTOD=57.29577951
      FOURPI=12.56637062
C
C
      READ (5,701) NCARDS
      REWIND 12
      WRITE (12) NCARDS
      WRITE(6,702)
      DO 10 J=1,NCARDS
      READ(5,703) HEAD
      WRITE (12) HEAD
   10 WRITE(6,704) HEAD
      WRITE(6,705)
C
C     SPECIFY FLIGHT CONDITIONS
C
      READ (5,706) ALFAC,FMACH
      WRITE (6,707) ALFAC,FMACH
      FMCHSQ=FMACH*FMACH
      BETASQ=FMCHSQ-1.0
      BETAO=SQRT(BETASQ)
      ALFACR=ALFAC*DTOR
C
C     SPECIFY AIRCRAFT COMPONENTS TO BE INCLUDED
C
      READ(5,701) NFU,NPY,NRACK,NSTRS
      IF(NSTRS.LT.8) GO TO 2
      WRITE (6,754) NSTRS
      STOP
C
C     INPUT FUSELAGE DATA IF FUSELAGE IS PRESENT
C
    2 NBIP=0
C           NFU=1, CIRCULAR BODY OPTION USING LINE SOURCES
C           NFU=2, USE GENERAL SOURCE PANELING METHOD FOR BODY
      IF (NFU.EQ.1) CALL FUSEIO
      IF (NFU.EQ.2) CALL WDYBDY
C
C     INPUT WING DATA
C
      READ(5,706) XBWOC,ZBWO,WIC
      WRITE(6,712)
      WRITE(6,713) XBWOC,ZBWO,WIC
      WICR = WIC*DTOR
      CWICR = COS(WICR)
      SWICR = SIN(WICR)
      CALL SWNGIN
      TIPY=SSPAN+Y(1)
      TLRNC=(TIPY*15E-5)**2
C
```

Figure A-1(a)

```
C     LAY OUT CONSTANT U-VELOCITY PANELS ON WING
C
      CALL WLYOUT
C
C     INPUT WING THICKNESS DATA
C
      CALL WITHIN
C
C     INPUT PYLON LAYOUT INFORMATION IF PYLON IS PRESENT
C
      MP=0
      MPS=0
      IF (NPY .EQ. 0) GO TO 20
      CALL PLYOUT
      CENTER=YPL.EQ.0.0
C
C     INPUT PYLON THICKNESS DATA
C
      CALL PYTHIN
   20 CONTINUE
      NTHP=MS+MPS
C
C     OUTPUT THICKNESS DISTRIBUTION
C
      CALL THKOUT
C
C     LAY OUT THICKNESS PANELS ON WING AND PYLON IF PRESENT
C
      CALL THKLYT
C
C     LAY OUT CONSTANT U-VELOCITY PANELS ON FUSELAGE IF PRESENT
C
      N1P=NPANLS+1
      N2=NPANLS+MP
      N2P=N2+1
      IF (NFU.EQ.1) CALL BLYOUT
      IF (NFU.EQ.2) CALL BLYOT2
      NPTOT=N2+NBIP
      NPTOTP=NPTOT+1
C
C     INPUT RACK DATA IF PRESENT
C
      IF (NRACK.EQ.0) GO TO 215
      CALL RACKIO
C
C     LOCATE RACK IN FUSELAGE COORDINATE SYSTEM
C
      SRIBCR=SIN(RIBCR)
      CRIBCR=COS(RIBCR)
      XBRO=XBWOC+XWROC*CWICR+ZWRO*SWICR
      YBRO=YWRO
      ZBRO=ZBWO -XWROC*SWICR+ZWRO*CWICR
  215 CONTINUE
C
C     INPUT STORE DATA IF PRESENT
C
      IF (NSTRS .EQ. 0) GO TO 30
      CALL STORIO
C
C     LOCATE STORE IN FUSELAGE COORDINATE SYSTEM
C
      DO 35 N=1,NSTRS
      SSIBCR(N)=SIN(SIBCR(N))
      CSIBCR(N)=COS(SIBCR(N))
      XBSO(N)=XBWOC+XWSOC(N)*CWICR+ZWSO(N)*SWICR
      YBSO(N)=YWSO(N)
   35 ZBSO(N)=ZBWO -XWSOC(N)*SWICR+ZWSO(N)*CWICR
   30 CONTINUE
C
C     SET UP COEFFICIENT MATRIX AND RIGHT HAND SIDE OF CONSTANT
C          U-VELOCITY EQUATION.  ARRANGE IN FVN ARRAY IN AUGMENTED
C          MATRIX FORM.
C
      CALL OPRHS
      CALL OPCOEF
      DO 40 I=1,NPTOT
      FVN(I,NPTOTP)=CIR(I)
   40 CONTINUE
C
C     SOLVE LINEAR SYSTEM
C
      CALL INVER1(FVN,1,NPTOT,200,  1)
      DO 43 J=1,NPTOT
      DELTP(J)=FVN(J,NPTOTP)
   43 CONTINUE
C
C
C     OUTPUT CONTROL POINT COORDINATES, INTERFERENCE VELOCITIES, INPUT
C          TWIST AND CAMBER ANGLES, AND U-VELOCITY PANEL SINGULARITY
C          STRENGTHS
C
C     WING PANELS
C
      WRITE (6,740)
      WRITE (6,741)
      DO 100 J=1,NPANLS
      NROW=(J-1)/NCW+1
      NPAN=J-NCW*(NROW-1)
      UPVINF=PI*DELTP(J)
  100 WRITE (6,742) NROW,NPAN,XCPT(J),YCPT(J),ZCPT(J),UEI(J),VEI(J),
     1WEI(J),ALPHAL(J),UPVINF
C
C     PYLON PANELS
C
      IF (NPY.EQ.0) GO TO 110
      WRITE (6,743)
      DO 101 J=N1P,N2
      NROW=(J-NPANLS-1)/NCP+1
      NPAN=J-NPANLS-NCP*(NROW-1)
      UPVINF=PI*DELTP(J)
  101 WRITE (6,744) NROW,NPAN,XCPT(J),YCPT(J),ZCPT(J),VEI(J),UPVINF
C
C     FUSELAGE PANELS
C
  110 IF (NFU.EQ.0) GO TO 120
      WRITE (6,745)
      DO 102 J=N2P,NPTOT
      NROW=(J-N2-1)/NBD+1
      NPAN=J-N2-NBD*(NROW-1)
      UPVINF=PI*DELTP(J)
  102 WRITE (6,746) NROW,NPAN,XCPT(J),YCPT(J),ZCPT(J),VEI(J),WEI(J),
     1 UPVINF
  120 CONTINUE
C
C     CALL ROUTINE TO FORM DATASETS FOR INPUT INTO
C     TRAJECTORY PROGRAM
C
      CALL NULYT
C
C     CALL ROUTINE TO WRITE FILE FOR INPUT INTO TRAJECTORY PROGRAM
C
      CALL WRFILE
      STOP
      END



      SUBROUTINE ASECTN (Y,Z,NRAD,AREA)
C
C  ROUTINE TO COMPUTE CROSS SECTIONAL AREA ENCLOSED BY ARBITRARY
C  Y AND Z SHAPE.
C
      DIMENSION Y(NRAD),Z(NRAD)
      COMMON /DIMENS/ DIM(130)
      EQUIVALENCE (DIM(38),IXZSYM)
      AREA = 0.0
      DO 100 I=2,NRAD
```

Figure A-1(b)

```
      IM1 = I-1
100   AREA = AREA+0.5*(Y(IM1)*Z(I)-Y(I)*Z(IM1))
      IF (IAZSYM.GE.0) AREA=AREA*2.0
      RETURN
      END



      SUBROUTINE BDYGEN(NXBODY,RADIUS,BODYL,NSEG,XEND,SCOEF,T,TC,TA,
     1 ALPHA,RPBODY,LBODY,SUMK,SUMKD)
C
C SUBROUTINE TO CALCULATE SUPERSONIC LINE SOURCES AND DOUBLETS TO GIVE A
C REQUIRED BODY SHAPE AND ANGLE OF ATTACK.
C
      DIMENSION XBODY(101),RBODY(101),RPBODY(101),DRDX(101),T(100)
      DIMENSION TC(100),TA(101)
      DIMENSION A(100),XF(100),RF(100)
      DIMENSION XEND(1)
C
      COMMON /FLOW/ALFACK,DUM1,MACH,DUM2(2),BETA,BETASQ,MACHSQ
      COMMON/SRCE/XFIELD,RFIELD,BSQ,X2,     U,V,VT
C
      REAL MACH,MACHSQ,LBODY
C
  700 FORMAT(/// 5X,106H***** AT BASE OF BODY RADIAL DISTANCE TO MACH CO
     1NE EMANATING FROM BODY NOSE IS LESS THAN MAXIMUM BODY RADIUS/13X,
     237HCHECK BODY INPUT DATA AND MACH NUMBER,60X,12H** BDYGEN **)
  701 FORMAT(/35X,10HBODY SHAPE,40X,11HSINGULARITY,5X,6HSOURCE,9X,
     17HDOUBLET/87X,6HORIGIN,7X,9HCONSTANTS,6X,9HCONSTANTS/24X,5HX, FT,
     210X,5HR, FT,10X,5HDR/DX,20X,1HN,7X,8HX(N), FT,8X,4HK(N),11X,
     35HKD(N))
  702 FORMAT(15X,3(0PF15.5),15X,I5,F15.5,2(1PE15.5))
  703 FORMAT(///5X,69H***** MACH CONE-BODY MERIDIAN INTERSECTION NOT FOU
     1ND AFTER 100 TRIALS,36X,12H** BDYGEN **)
C
      LBODY=BODYL*XEND(NSEG)
      IF(BETA*RADIUS.LT.LBODY) GO TO 20
      WRITE (6,700)
      STOP
   20 N=NXBODY-1
      XBODY(1)=0.0
      XBODY(NXBODY)=LBODY
C
C SETUP OF POINTS ON BODY AXIS BY DIVIDING BODY LENGTH INTO N  EQUAL
C    LENGTH SEGMENTS
C
      FN=N
      DEL=LBODY/FN
      DO 33 I=2,N
   33 XBODY(I)=XBODY(I-1)+DEL
C
C CALCU  TION OF RADII AND RPBODY TO GO WITH THE AXIS POINTS
C
      OLBOD=1./BODYL
      DO 35 I=1,NXBODY
      XOL=XBODY(I)*OLBOD
      CALL SHAPE(XOL,NSEG,XEND,SCOEF,ROL,RPBODY(I))
   35 RBODY(I)=ROL*BODYL
C
C  CALCULATION OF DRDX AND RADIUS AT MIDPOINTS BETWEEN AXIS POINTS
C
      DO 36 I=1,N
      XF(I)=.5*(XBODY(I)+XBODY(I+1))
      XOL=XF(I)*OLBOD
      CALL SHAPE(XOL,NSEG,XEND,SCOEF,ROL,DRDX(I))
   36 RF(I)=ROL*BODYL
C
C THIS LOOP DETERMINES THE LOCATIONS OF THE ORIGINS OF THE CONICAL LINE
C SOURCES AND DOUBLETS.  THE ORIGINS ARE LOCATED AT THE TX(I).
C
      DO 10 I=1,NXBODY
   10 TX(I)=XBODY(I)-BETA*RBODY(I)
C
C     IF CONTROL POINTS LIE O TSI   THE MA H C NE WITH ITS ORIGIN AT THE
C          BODY NOSE, REVISE LAYOUT FOR BODY DEFINITION POINTS,ETC.
C
      IF (BETA*RF(1).LT.XF(1)) GO TO 199
C
C     DETERMINE X-STATION OF INTERSECTION OF BODY NOSE MACH CONE AND
C          BODY SURFACE
C
      DO 37 I=2,N
      IF ((BETA*RF(I)-XF(I)).LT.0.0) GO TO 38
   37 CONTINUE
   38 IAFT=I
      IBFR=I-1
      DELTX=0.1*(XF(IAFT)-XF(IBFR))
      ITRY=0
      XTRY=XF(IBFR)
   40 RMCNE=XTRY/BETA
      XOL=XTRY*OLBOD
      CALL SHAPE(XOL,NSEG,XEND,SCOEF,ROL,SLPE)
      ITRY=ITRY+1
      RI=ROL*BODYL
      ERR=RI-RMCNE
      IF (ITRY.GE.100) GO TO 42
      IF (ERR.LT.0.0) GO TO 39
      XTRY=XTRY+DELTX
      GO TO 40
   42 WRITE (6,703)
      STOP
   39 IF (ABS(ERR).LT.(0.01*RI)) GO TO 41
      XTRY=XTRY-DELTX
      DELTX=0.1*DELTX
      GO TO 40
   41 XI=XTRY
C
C     REVISE LAYOUT OF BODY DEFINITION POINTS,CONTROL POINTS, AND
C          ORIGINS OF LINE SINGULARITIES
C
      SBODY=LBODY-XI
      DEL=SBODY/FN
      XBODY(2)=XI+DEL
      DO 43 I=3,N
   43 XBODY(I)=XBODY(I-1)+DEL
      DO 44 I=1,NXBODY
      XOL=XBODY(I)*OLBOD
      CALL SHAPE (XOL,NSEG,XEND,SCOEF,ROL,RPBODY(I))
      RBODY(I)=ROL*BODYL
   44 CONTINUE
      XF(1)=0.5*(XI+XBODY(2))
      XOL=XF(1)*OLBOD
      CALL SHAPE(XOL,NSEG,XEND,SCOEF,RO ,DRDX(1))
      RF(1)=ROL*BODYL
      DO45 I=2,N
      XF(I)=0.5*(XBODY(I)+XBODY(I+1))
      XOL=XF(I)*OLBOD
      CALL SHAPE(XOL,NSEG,XEND,SCOEF,ROL,DRDX(I))
      RF(I)=ROL*BODYL
   45 CONTINUE
      TX(1)=0.0
      DO 46 I=2,NXBODY
   46 TX(I)=XBODY(I)-BETA*RBODY(I)
  199 CONTINUE
C
C     DETERMINATION OF SOU-CE STRENGTHS AT CONTROL POINTS MIDWAY BETWEEN
C      BODY DEFINITION POINTS.
C
C CALCULATION OF THE FIRST SOURCE STRENGTH.
C
      XFIELD=XF(1)
      RFIELD=RF(1)
      SLOPE=DRDX(1)
      CALL SOURCE(TX(1))
      A(1)=V-SLOPE*U
      T(1)=DRDX(1)/A(1)
C
```

Figure A-1(c)

```
C CALCULATION OF THE REST OF SOURCE STRENGTHS
C
      DO 210 I=2,N
      XFIELD=XF(I)
      RFIELD=RF(I)
      SLOPE=DRDX(I)
      DO 205 J=1,I
      CALL SOURCE(TX(J))
  205 A(J)=V-SLOPE*U
      SUM=0.
      IM1=I-1
      DO 201 J=1,IM1
  201 SUM=T(J)*A(   J)+SUM
  210 T(I)=(DRDX(I)-SUM)/A(I)
      T(NXBODY)=0.
C
C     DETERMINATION OF DOUBLET STRENGTHS AT CONTROL POINTS MIDWAY
C      BETWEEN BODY DEFINITION POINTS
C
C CALCULATION OF THE FIRST DOUBLET STRENGTH.
C
      XFIELD=XF(1)
      RFIELD=RF(1)
      SLOPE=DRDX(1)
      CALL DOUBLT(TX(1))
      A(1)=SLOPE*U-V
      TC(1)=ALPHA/A(1)
C
C CALCULATION OF THE REST OF THE DOUBLET STRENGTHS
C
      DO 215 I=2,N
      XFIELD=XF(I)
      RFIELD=RF(I)
      SLOPE=DRDX(I)
      DO 212 J=1,I
      CALL DOUBLT(TX(J))
  212 A(   J)=SLOPE*U-V
      SUM=0.
      IM1=I-1
      DO 203 J=1,IM1
  203 SUM=TC(J)*A(   J)+SUM
  215 TC(I)=(ALPHA-SUM)/A(I)
      TC(NXBODY)=0.
C
C PRINT OUT OF BODY CHARACTERISTICS
C
      WRITE (6,701)
      DO 800 I=1,NXBODY
      II=I-1
  800 WRITE(6,702)    XBODY(I),RBODY(I),RPBODY(I),II,TX(I),T(I),TC(I)
C
C     CALCULATE SUM OF T(J) AND TC(J)
C
      SUMK=0.0
      SUMKD=0.0
      DO 801 J=1,N
      SUMK=SUMK+T(J)
  801 SUMKD=SUMKD+TC(J)
      RETURN
      END


      SUBROUTINE BLYOUT
C
C     SUBROUTINE TO LAY OUT FUSELAGE CONSTANT U-VELOCITY PANELS
C
      COMMON/BPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1     SANGW
      COMMON /CONSTS/PI,PI2,DTOR,RTOD,FOURPI
      COMMON /FSGEOM/FRMAX,SSPAN,    FLTHC,BODYPL
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,NIP,N2,N2P,NPTOT
      COMMON/WGEOM/XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1      PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR
      COMMON/WPGEOM/XRF(200),XRR(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2     SNPHI(200),CSPHI(200)
C
      DATA FAC/0.95/
C
C     FUNCTION STATEMENTS WHICH TRANSFORM BODY PANEL
C        COORDINATES TO WING COORDINATE SYSTEM.
C
      YWING(YPAN)=YPAN*CS2-FRMAX*CS
      ZWING(YPAN)=YPAN*SN2-FRMAX*SN-ZBWO
C
C     LAY OUT FUSELAGE CONSTANT U-VELOCITY PANELS
C
      ANCWB=NCWB
      DX=BODYPL/ANCWB
      NBD=NBDCR1+NBDCR2
      NBIP=NBD*NCWB
      SANGW=ZBWO/FRMAX
      ANGW=ASIN(SANGW)
      IF(ABS(ANGW-PI2).LE..0017) GO TO 270
      IF(ABS(ANGW+PI2).LE..0017) GO TO 280
C
C     WING IS NOT TANGENT TO FUSELAGE
C
      IL=1
      IU=NBDCR2
      ANGB=-PI
      AN2=NBDCR2
      THTS=(PI2-ANGW)/AN2
      THT=-PI2
      ITIMES=1
  210 WBIP=2.0*FRMAX*SIN(THTS/2.0)
      CPY=0.5*WBIP
      AI=0.0
      DO 260 I=IL,IU
      XFBIP=0.0
      AI=AI+1.0
      ANG=ANGB+(AI-0.5)*THTS
      SN=SIN(THT)
      CS=COS(THT)
      SN2=SIN(ANG)
      CS2=COS(ANG)
      DO 255 J=I,NBIP,NBD
      K=J+N2
      SNT2(J)=SN2
      CST2(J)=CS2
C
C     TRANSFORM CONTROL POINT Y AND Z COORDINATES INTO WING SYSTEM
C
      YCPT(K)=YWING(CPY)
      ZCPT(K)=ZWING(CPY)
C
C     TRANSFORM CORNER POINT Y AND Z COORDINATES INTO WING SYSTEM
C
      YLC(K)=YWING(0.)
      ZLC(K)=ZWING(0.)
      YRC(K)=YWING(WBIP)
      ZRC(K)=ZWING(WBIP)
      XLF(K)=XFBIP
      XRF(K)=XFBIP
      XFBIP=XFBIP-DX
      XLB(K)=XFBIP
      XRB(K)=XFBIP
C
C     CONTROL POINT X-COORDINATE
C
      XCPT(K)=XFBIP+(1.0-FAC)*DX
C
C     POLAR ANGLE OF THIS PANEL, DEGREES
C
      THTI(J)=(PI-THT)*RTOD
```

Figure A-i(d)

```
  255 CONTINUE
      THT=THT+THTS
  260 CONTINUE
      IF(ITIMES.EQ.2) GO TO 300
      IL=NBDCR2+1
      IU=NBD
      AN1=NBDCR1
      ANGB=-ANGW-PI2
      THTS=(PI2+ANGW)/AN1
      THT=-ANGW
      ITIMES=2
      GO TO 210
C
C     WING TANGENT TO BOTTOM OF FUSELAGE
C
  270 IL=1
      IU=NBDCR1
      AN1=NBDCR1
      ANGB=-PI
      THTS=PI/AN1
      THT=-PI2
      ITIMES=2
      GO TO 210
C
C     WING TANGENT TO TOP OF FUSELAGE
C
  280 IL=1
      IU=NBDCR2
      ANGB=-PI
      AN2=NBDCR2
      THTS=PI/AN2
      THT=-PI2
      ITIMES=2
      GO TO 210
  300 CONTINUE
      RETURN
      END


      SUBROUTINE BLYOT2
C
C  ROUTINE TO LAYOUT THE FUSELAGE U-VELOCITY PANELS FOR THE
C  GENERAL BODY SHAPE.  THE SECTION GEOMETRY IS OBTAINED BY
C  INTERPOLATION AT THE USER SPECIFIED BODY X-STATION IN THE
C  ARBITRARY FUSELAGE GEOMETRY DEFINITION.  THE PANELING OF THE BODY
C  WILL BE PERFORMED BY THE USER SO AS TO GUARANTEE THE MATCHING OF
C  PANEL EDGES WITH THE WING ROOT CHORD.
C  THT IS POLAR ANGLE DEFINING ORIGIN OF PANEL COORDINATE SYSTEM
C  RELATIVE TO FUSELAGE CENTER.  POSITIVE IN CLOCKWISE ROTATION FROM
C  NEGATIVE YB AXIS.
C
      COMMON /BIPYZ / XBIP,NYZBIP,YCB(33),ZCB(33)
      COMMON /BPGEOM/ SNT2(200),CST2(200),ZLC(200),ZRC(200),THT1(200),
     1      SANGW
      COMMON /CONSTS/ PI,PI2,DTOR,RTOD,FOURPI
      COMMON /FSGEOM/ FRMAX,SSPAN,    FLTHC,BODYPL
      COMMON /INDEX / NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP
     1    ,NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON /WGEOM / XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1      PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR
      COMMON /WPGEOM/ XRF(200),XRB(200),XLF(200),XLR(200),YRC(200),
     1    YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2      SNPHI(200),CSPHI(200)
C
      DATA FAC/0.95/
      ANCWB=NCWB
      DX=BODYPL/ANCWB
      NBD = NYZBIP-1
      NBIP=NBD*NCWB
C
C  LOOP THROUGH NUMBER OF PANELS AROUND BODY
C
      DO 120 I=1,NBD
      XFBIP = 0.0
      IP1 = I+1
      YI = -YCB(IP1)
      YIM1 = -YCB(I)
      ZI = -ZCB(IP1)
      ZIM1 = -ZCB(I)
      THT = ATAN2(-ZIM1,-YIM1)
      DY = YI-YIM1
      DZ = ZI-ZIM1
      WBIP = SQRT(DY*DY+DZ*DZ)
      SN2 = DZ/WBIP
      CS2 = DY/WBIP
C
C  LOOP OVER PANELS IN AXIAL DIRECTION WHICH HAVE SAME GEOMETRY
C
      DO 110 J=1,NBIP,NBD
      K = J+N2
      SNT2(J) = SN2
      CST2(J) = CS2
C
C  POLAR ANGLE OF THIS PANEL,  DEGREES.
      THT1(J) = (PI-THT)*RTOD
C
C
C  CONTROL POINT X,Y,Z COORDINATES IN WING SYSTEM
C
      XCPT(K) = XFBIP-FAC*DX
      YCPT(K) = 0.5*(YI+YIM1)
      ZCPT(K) = 0.5*(ZI+ZIM1)
C
C  CORNER POINT X,Y,Z COORDINATES IN WING SYSTEM
C
      YLC(K) = YIM1
      ZLC(K) = ZIM1
      YRC(K) = YI
      ZRC(K) = ZI
      XLF(K) = XFBIP
      XRF(K) = XFBIP
      XFBIP = XFBIP-DX
      XLB(K) = XFBIP
      XRB(K) = XFBIP
  110 CONTINUE
  120 CONTINUE
      RETURN
      END


      SUBROUTINE BODPAN
C
C   REVISE AXIAL SPACING ON BODY (FUSELAGE) AND COMPUTE NEW PANEL
C   GEOMETRY
C
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSRAD(51)
     *    ,FUSAZ(51),XJ(51),PHIK(33)
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BOOL,IZ1(12)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *    ,IAR,IAN,IUB,IGB(7),IVR,IU,IV,IW,IVA,IWA,ICP,IPHI,IYR
     *    ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IA0,ID0,ISK0,IZ2(4),NRING,IROW(50)
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1    ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON            A(12000)
C
      DATA DEGRD /57.29577951/
C
      REWIND 8
C
```

Figure A-1(c)

AD-A099 331 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 20/4
PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLU--ETC(U)
NOV 80 J MULLEN, F K GOODWIN, M F DILLENIUS F33615-76-C-3077
UNCLASSIFIED NEAR-TR-210-VOL-3 AFWAL-TR-80-3032-VOL-3 NL

2 OF 3
AD A09933

```
C   CALCULATE COORDINATES OF PANEL CORNERS  FOR EACH SEGMENT
C             JP = INDEX OF PANEL NUMBER (TOTAL=NBODY)
C
      IF (IPRT(2).GT.0) WRITE (6,220)
      NRING = 0
      JP=0
C             BLANK COMMON ARRAY LOCATIONS
      IXB = 1
      IXJ = 1
      IXPT = 1
      IYPT = NBODY+IXPT
      IZPT = NBODY+IYPT
      ITH  = NBODY+IZPT
      IDEL = NBODY+ITH
      IAR  = NBODY+IDEL
      IXC0 = NBODY+IAR
      IYC0 = IXC0+KX
      IZC0 = IYC0+KXKR
      DO 120 IFU=1,NFUS
      NYZ = NFORX(IFU)*KRADX(IFU)
      IF (NFUS.GT.1) CALL IOREAD (8,A(IYB),2*NYZ)
      IZB = IYB+NYZ
      IXC(IFU) = IXC0
      IYC(IFU) = IYC0
      IZC(IFU) = IZC0
C                                    YB      ZB      XPT  XC
      CALL BODPN2 (JP,XFUS(IXB),XJ(IXJ),A(IYB),A(IZB),A(1),A(IXC0)
     *   ,A(IYC0),A(IZC0),NFORX(IFU),KRADX(IFU),KFORX(IFU),NBODY,IPRT)
C         YC        ZC
      IXB = IXB+NFORX(IFU)
      IXJ = IXJ+KFORX(IFU)
      IXC0 = IXC0+KFORX(IFU)
      IYC0 = IYC0+KFORX(IFU)*KRADX(IFU)
      IZC0 = IZC0+KFORX(IFU)*KRADX(IFU)
120   CONTINUE
      REWIND 8
C
C  PRINT PANEL CONTROL POINTS, INCLINATION ANGLES, AND AREAS
C
      IF (IPRT(2).LT.1) GO TO 140
      WRITE (6,250)
      NB6 = NBODY*6
      NB3 = NBODY*3
      DO 130 JP=1,NBODY
      JDEG = NB3+JP
      TDEG = A(JDEG)*DEGRD
      JDEG = JDEG+NBODY
      DDEG = A(JDEG)*DEGRD
130   WRITE(6,270) JP,(A(I),I=JP,NB6,NBODY),TDEG,DDEG
C
C   STORE BODY GEOMETRY ON TAPE 7
C
140   NTAP7 = NG
      CALL IOWRIT (7,A,NTAP7)
      RETURN
220   FORMAT (1H1,9X,35HBODY PANEL CORNER POINT COORDINATES
     * 45X,12H** BODPAN **,///10X,5H1 AND
     1 60H 3 INDICATE BODY PANEL LEADING-EDGE POINTS, 2 AND 4 INDICATE
     2 21H TRAILING-EDGE POINTS/5X,5HPANEL,4(8X,1HX,8X,1HY,8X,1HZ)/
     3 19X,3(1H1,8X),3(1H2,8X),3(1H3,8X),3(1H4,8X)/)
250   FORMAT (1H1,38H BODY PANEL CENTROID POINT COORDINATES
     1   30H, INCLINATION ANGLES AND AREAS
     2 //4X,5HPOINT,7X,1HX,10X,1HY,10X,1HZ,7X,5HTHETA,6X,5HDELTA
     3   ,7X,4HAREA,6X,5HTHETA,6X,5HDELTA/6X,3(9X,2HCP)
     4   ,2(9X,3HRAD),11X,2(8X,3HDEG)/)
270   FORMAT (I8,1X,6F11.5,F11.3,F11.5)
      END


      SUBROUTINE BODPN2 (JP,XB,XJ,YB,ZB,XYZ,XC,YC,ZC,
     *   NFUSOR,KRAD,KFUSOR,NBODY,IPRT)
C
C  ROUTINE TO  PERFORM THE INTERPOLATION ON X IN BODY GEOMETRY
C  AND COMPUTE PANEL COORDINATES AND OTHER PANEL GEOMETRIC PROPERTIES
C  ARRAYS$ XFUS,XJ,YB,ZB,XC,YC,ZC PERTAIN ONLY TO A GIVEN SEGMENT
C           XYZ = (XPT,YPT,ZPT,THET,DELTA,AREA) PANEL PROPERTIES
C                 STORED BY PANEL.
C    J = INDEX OF AXIAL STATION IN XC FOR A GIVEN SEGMENT
C   JP = PANEL NUMBER
C
       DIMENSION XB(NFUSOR),XJ(KFUSOR),YB(NFUSOR,KRAD),ZB(NFUSOR,KRAD)
     *   ,XYZ(NBODY,6),XC(KFUSOR),YC(KFUSOR,KRAD),ZC(KFUSOR,KRAD)
     *   ,IPRT(5)
       COMMON /DIMENS/ IZDIM(79),NRING,IROW(50)
C
       XIN(X1,Y1,X2,Y2,Y)=X1+(X2-X1)*(Y-Y1)/(Y2-Y1)
C
C  INITIALIZE COORDINATES EQUAL TO SEGMENT LEADING EDGE
C
       XC(1)=XB(1)
60     DO 70 K=1,KRAD
       YC(1,K)=YB(1,K)
       ZC(1,K)=ZB(1,K)
70     CONTINUE
C
C  INTERPOLATE FOR REMAINING PANEL COORDINATES  -----------------------
       DO 110 J=2,KFUSOR
       JM1=J-1
       NRING = NRING+1
       IROW(NRING) = KRAD-1
C
C  FIND X-STATION - SEARCH FOR EXTERNAL GEOMETRY X-STATION(XB) BETWEEN
C  WHICH XJ LIES
C
       DO 100 M=2,NFUSOR
       IF (XB(M).LT.XJ(J)) GO TO 100
       MM1=M-1
       DO 90 K=1,KRAD
       XC(J)=XJ(J)
       YC(J,K)=XIN(YB(MM1,K),XB(MM1),YB(M,K),XB(M),XJ(J))
       ZC(J,K)=XIN(ZB(MM1,K),XB(MM1),ZB(M,K),XB(M),XJ(J))
       IF (K.EQ.1) GO TO 90
       KM1=K-1
       JP=JP+1
C
C   CALCULATE PANEL INCLINATION AND CENTROID
C
       CALL PANEL (J,K,JP,XYZ,NBODY,XC,YC,ZC,KFUSOR)
C
C  PRINT BODY PANEL COORDINATES
       IF (IPRT(2).GT.0)
     *    WRITE (6,240) JP,XC(JM1),YC(JM1,KM1),ZC(JM1,KM1),XC(J),
     1    YC(J,KM1),ZC(J,KM1),XC(JM1),YC(JM1,K),ZC(JM1,K),
     2    XC(J),YC(J,K),ZC(J,K)
90        CONTINUE
          GO TO 110
100     CONTINUE
110   CONTINUE
150   RETURN
C
240   FORMAT (I8,4X,12F9.5)
      END


      SUBROUTINE BODVEL (XPT,YPT,ZPT,THET,DELTA,AN,UB
     *    ,XC,YC,ZC,KFUSOR,IFU,IXZSYM)
C
C  COMPUTE THE THREE COMPONENTS OF VELOCITY INDUCED AT SPECIFIED
C  CONTROL POINTS BY THE BODY PANELS FOR A UNIT STRENGTH.
C  THESE ARE USED TO GENERATE THE COEFFICIENT MATRICES USED TO
C  COMPUTE THE PANEL STRENGTHS AND RESULTING COMPONENT VELOCITIES
C  ON EACH PANEL.
C
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIV0,NINALK,BTINLT,YCPI
```

Figure A-1(f)

```
     *    ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BLKPAN/ COST,SINT,XBTJ,YBTJ,ZBTJ,XC1,YC1,ZC1
     *    ,XPTI,YPTI,ZPTI,COSTI,SINTI,COSO,SIND,LZERO
      COMMON /BODCOM/ AMACH,TANO,CA,XCOP(4),YCOR(4),ZCOR(4)
     1    ,XI,YI,ZI,XJ,ZJ,BETA0,BETAL,SUBSON,SUPERS,J
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XMLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /DIMENS/ IZDIM(79),NRING,IROW(50)
C
      DIMENSION XPT(1),YPT(1),ZPT(1),THET(1),DELTA(1),AN(1),UB(3,1)
     *    ,XC(KFUSOR),YC(KFUSOR,1),ZC(KFUSOR,1)
      LOGICAL LZERO,SUBSON,SUPERS
      LOGICAL INLTST,INLET,OPEN
      EXTERNAL INLTST
      LZERO = .FALSE.
C             FOR AIC VELOCITY CALCULATION, CONSIDER ONLY INLET PANELS
      NTINL = NINLET*NINBLK
C
C  JB IS THE INDEX OF THE PANEL RING
C  IB IS THE INDEX OF THE RING ON WHICH THE CONTROL POINT LIES
C  JG IS STARTING OFFSET OF GB(J) (COLUMN NUMBER OF BLOCK)
C  IO IS STARTING OFFSET OF VB(I) (ROW NUMBER OF BLOCK)
C  I IS THE INDEX OF THE CONTROL POINT
C  J IS THE INDEX OF THE INFLUENCING PANEL
C
C             LOOP THROUGH NUMBER OF RINGS IN SEGMENT OF INFLUENCING
C             PANEL BLOCKS
      JS = 1
      IF (IFU.LE.1) JB=0
      IF (IFU.LE.1) JG=0
      DO 110 L=2,KFUSOR
      JB = JB+1
      JR = IROW(JB)
      IF (SUPERS) JS=JB
C             LOOP THROUGH ALL RINGS FOR CONTROL POINT BLOCKS
      IO = 0
      DO 100 IB=JS,NRING
      IF (IB.EQ.JB) IO =JG
      IR = IROW(IB)
C             DEFINE SUBMATRIX A(IB,JB)
C             FOR SUBMATRIX - LOOP ROWWISE THROUGH PANELS ON A RING
      IJ = 0
      DO 60 N=1,JR
      J = JG+N
      XBTJ = XPT(J)
      YBTJ = YPT(J)
      ZBTJ = ZPT(J)
      TAND=TAN(DELTA(J))
      COST=COS(THET(J))
      SINT=SIN(THET(J))
C
C  FOR INLET, SET TO GREATER OF BTINLT OR PANEL SLOPE
C
      BETAL = BETA0
      IF (.NOT.INLTST(J,OPEN)) GO TO 30
      BETAL = AMIN1(BTINLT,BETA0)
      IF (BTINLT*TANO.GT.1.) BETAL=0.99/TANO
C
30    LM1 = L-1
      NP1 = N+1
      XC1=XC(LM1)
      YC1=YC(LM1,NP1)
      ZC1=ZC(LM1,NP1)
C
C  CALCULATION OF PANEL CORNER POINTS IN PANEL COORDINATE SYSTEM
C             K= 1        LM1= L-1        NP1= N+1
C                2             L               N+1
C                3             L-1             N
C                4             L               N
C
      XCOR(1)=0.
      YCOR(1)=0.
      ZCOR(1)=0.
      XCOR(3)=0.
      XCOR(2)=XC(L)-XC1
      XCOR(4)=XCOR(2)
      DO 40 K=2,4
      LM1 = L
      NP1 = N+1
      IF (K.GE.3) NP1=N
      IF (K.EQ.3) LM1=L-1
      DELY=YC(LM1,NP1)-YC1
      DELZ=ZC(LM1,NP1)-ZC1
      YCOR(K)=DELY*COST+DELZ*SINT
      ZCOR(K)=DELZ*COST-DELY*SINT
40    CONTINUE
      CX=XCOR(2)
C             FOR SUBMATRIX - LOOP COLUMNWISE THROUGH CONTROL POINTS
      DO 50 IP=1,IR
      IJ = IJ+1
      I = IO+IP
      XPTI = XPT(I)
      YPTI = YPT(I)
      ZPTI = ZPT(I)
      COSTI = COS(THET(I))
      SINTI = SIN(THET(I))
      COSD = COS(DELTA(I))
      SIND = SIN(DELTA(I))
      CALL PANVEL (UB(1,IJ),UB(2,IJ),UB(3,IJ),AN(IJ),IXZSYM)
50    CONTINUE
60    CONTINUE
C
C  SUBMATRICES ARE GENERATED AND STORED IN COLUMN ORDER CORRESPONDING
C  TO INFLUENCE OF A RING OF PANELS ON ALL FOLLOWING CONTROL POINTS.
C  STORE SUBMATRICES ON DISK FOR LOWER TRIANGLE BY RING.
C         TAPE8  CONTAINS U,V,W    COMPONENTS
C         TAPE9  CONTAINS AN       NORMAL
C
      NB = 3*IJ
      IF (IUVW.EQ.0) CALL IOWRIT (8,UB,NB)
C
C  IF DIAGONAL BLOCKS  DECOMPOSE MATRIX INTO L*U TRIANGULAR MATRIX
C  FORM FOR LATER SOLUTION.
C
      IF (IB.EQ.JB) CALL PAS001(AN,JR,NER,JR)
      IF (NER.GT.0) STOP 100
      CALL IOWRIT (9,AN,IJ)
C
100   IO = IO+IR
110   JG = JG+JR
      RETURN
      END


      SUBROUTINE BSHOCK(RINIT,PHISHK,THSHK)
C
C  THIS SUBROUTINE CALCULATES THE SHAPE OF THE SHOCK WAVE PRODUCED
C  BY AN ARBITRARY BODY AT ANGLE OF ATTACK, ALPHAC, ALONG A GIVEN
C  MERIDIAN, THETA.
C  INPUTS
C    RINIT  = RADIUS AT WHICH SHOCK SHAPE SEARCH IS INITIATED.
C             APPROXIMATELY EQUAL TO MAXIMUM BODY RADIUS.
C    PHISHK = NSHOCK=0, ANGLE AT WHICH SHOCK TRAVERSE IS MADE
C             .GT.0,MAX MERIDIANL ANGLE TO WHICH SHOCK IS CALCULATED
C             ON EVEN SPACING
C    THSHK  = LIMITING SHOCK WAVE ANGLE AT NOSE.
C             MEASURED FROM X-AXIS
C
C  BLANK COMMON USAGE
C  (1) INITIALIZE GEOMETRY AND SOURCE STRENGTHS AFTER SOLUTION
C        XPT,YPT,ZPT,THET,DELTA,AREA,XC,YC,ZC,GB,U,V,W,CP,...
C  (2) FOR FIELD VELOCITY CALCULATION
C        XPT,....GB,XS,YS,ZS,US,VS,WS,SVN,LSKP,...
C
```

Figure A-1(g)

```
      LOGICAL SMALL,SUBSON,SUPERS,IPGTO
      COMMON /BODCOM/ AMACH,TAND,CA,XCOR(4),YCOR(4),ZCOR(4)
     1    ,XI,YI,ZI,XJ,ZJ,BETA0,BETAL,SUBSON,SUPERS,J
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XMLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,DBETA
     *    ,EALPHA,CNU0,CNU2,XSHLDR,SHK(3),XSHK(100),RSHK(100)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXRW,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IAC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NADIM,NG,IXPT,IYPT,IZPT,ITM,IDEL,NTAP7
     *    ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IY9
     *    ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IA0,ID0,ISR0,IZZ(4),NWING,IROW(50)
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1    ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON             A(12000)
C
      DATA DGRAD/57.2957795/
C
C  DOUBLE ANGLE TRIG IDENTITY FOR TAN(A+B)
C
      TANAPB(TANA,TANB) = (TANA+TANB)/(1.0-TANA*TANB)
C
701   FORMAT(///30X,31HMODIFIED SHOCK LOCATIONS PHIS =,F8.3,
     *    41X,12H** BSHOCK **,
     *    //1HX,1HN,5X,1HX,9X,1HR,9X,1HU,9X,1HW,9X,2HML,7X,4HBETA
     *    ,4X,5HDX/DR,7X,3HDNU,5X,5HUVWSQ)
702   FORMAT(10X,I5,2F10.3,7F10.5)
707   FORMAT(10F7.3)
C
      DTOR = 1./DGRAD
      IPGTO = IPRT(3).GT.0
C
C  INITIALIZATION -----------------------------------------------------
C
      BETA0 = SQRT(XMACH*XMACH-1.)
      BETAL = BETA0
      XTNU = TAN(ASIN(1./XMACH))
      DXS = 0.2/FLOAT(MAX0(1,IFIX(30./REFL)))
      NFLD = IFIX(XSHLDR/(2.*DXS))+1
      NFLD0 = NFLD
      XMAX = 2.*REFL
C
C  DEFINE STARTING LOCATION OF ARRAYS IN BLANK COMMON
C
      IXS = IU
      IYS = IXS+NFLD
      IZS = IYS+NFLD
      IUS = IZS+NFLD
      IVS = IUS+NFLD
      IWS = IVS+NFLD
      ISVN = IWS+NFLD
      ISKP = ISVN+NFLD
      MASHK = ISKP+NFLD-1
C
C  CALCULATE ANGLES FOR SHOCK TRAVERSES
C    IF NSHOCK= (+) USE EVEN INCREMENT, (0) SINGLE TRAVERSE AT PHISHK
C               (-) READ ANGLES ALONG WHICH SHOCK SHAPE IS COMPUTED
C
      IF (NSHOCK) 70,80,50
C
C  (NSHOCK.GT.0) COMPUTE NSHOCK EVENLY SPACED SHOCK LOCATIONS
50    PHIS(1) = 0.
      IF (NSHOCK.EQ.1) GO TO 90
      DANGL = PHISHK/FLOAT(NSHOCK-1)
      DO 60 I=2,NSHOCK
60    PHIS(I) = PHIS(I-1)+DANGL
      GO TO 90
C
C  (NSHOCK.LT.0) READ NSHOCK ANGLES FROM INPUT
70    NSHOCK = IABS(NSHOCK)
      READ(5,707) (PHIS(I),I=1,NSHOCK)
      GO TO 90
C
C  (NSHOCK=0)     MAKE SINGLE TRAVERSE AT POLAR ANGLE PHISHK
80    NSHOCK = 1
      PHIS(1) = PHISHK
90    CONTINUE
      DO 200 IRAD=1,NSHOCK
      NFLD = NFLD0
      THETN(IRAD) = THSHK
      PHISI = PHIS(IRAD)
      NSHK(IRAD) = 1
      ISHK = MAXSHK*(IRAD-1)+1
      XSHK(ISHK)=0.0
      RSHK(ISHK)=0.0
C
C  MODIFY NOSE SHOCK ANGLE SHAPE TO INCLUDE ANGLE OF ATTACK VARIATION
C  REF% KRASNOV, AERODYNAMICS OF BODIES OF REVOLUTION, EQ. 7.40.
C  PSI  IS THE ANGLE OF THE POLAR ANGLE OF THE SHOCK SEARCH RELATIVE
C    TO THE INCIDENT FLOW.  AT ALPHA=0, IT IS MEASURED FROM THE
C    BOTTOM CENTERLINE.
C  BTNOSE IS TANGENT OF 90-NOSE SHOCK ANGLE PLUS ALPHA CORRECTION
C  DTHETN IS ANGLE OF ATTACK CORRECTION TO NOSE SHOCK ANGLE.
C
      THETNS = THSHK*DTOR
      ALP = ALPHAC*DTOR
      PSI = (PHISI-PHIR)*DTOR
      DTHETN = ALP*(1.-EALPHA)*COS(PSI)
C                   90 DEG.
      BTNOSE = TAN(1.5707963-THETNS+DTHETN)
      TANALP = TAN(ALP*COS(PSI))
      BETINF = TANAPB(BETA0,TANALP)
      FML = SQRT((TANAPB(BTNOSE,-TANALP))**2+1.)
C                INITIALIZE INCREMENTS IN R AND X FOR INTEGRATION
      SINT = SIN(PHISI/DGRAD)
      COST = COS(PHISI/DGRAD)
      DR = 0.5*RINIT
      R = 1.5*RINIT-DR
      BSAVE = BTNOSE
      ANGNU1 = (2.4495*ATAN(0.40825*BETA0)-ATAN(BETA0))*DGRAD
C
C  RADIAL LOCATION LOOP -------------------------------------------
C
      UWR = 0.0
      VWSAV = 0.0
      IF (IPGTO) WRITE(6,701) PHISI
      IF (IPGTO) WRITE(6,702) NSHK(IRAD),XSHK(ISHK),RSHK(ISHK),UWR,VWSAV
     *    ,FML,BTNOSE
100   R = R+DR
      ISHK = ISHK+1
      Y = R*SINT
      Z = -R*COST
      X = R/XTNU-0.001*DXS
C
C  DEFINE LOCATION OF TRAVERSE TO FIND MAXIMUM CROSS FLOW VELOCITY
C  AFTER NOSE SHOCK.
      DO 110 I=1,NFLD
      IM1 = I-1
      JXS = IXS+IM1
      JYS = IYS+IM1
      JZS = IZS+IM1
      A(JXS) = X
      A(JYS) = Y
      A(JZS) = Z
110   X = X+DXS
C
C  COMPUTE VELOCITIES AT FIELD POINTS
C
      CALL FLDVEL (A(IXS),A(IYS),A(IZS),A(ISVN),A(ISKP)
     *    ,A(IUS),A(IVS),A(IWS),NFLD,NX,1)
C
C  SCAN CROSS FLOW VELOCITIES TO FIND MAXIMUM VALUE
C
      UWR = 1.0
      VWR = 0.0
      WWR = 0.0
```

Figure A-1(h)

```
      VWSAV = 0.0
      DO 150 I=1,NFLD
      IM1 = I-1
      JUS = IUS+IM1
      JVS = IVS+IM1
      JWS = IWS+IM1
      USI = A(JUS)
      VSI = A(JVS)
      WSI = A(JWS)
      VWMAG = 0.
      IF (VSI.EQ.0. .AND. WSI.EQ.0.) GO TO 140
      VWMAG = SQRT(VSI*VSI+WSI*WSI)
      IF (VWMAG.LE.VWSAV) GO TO 140
      VWSAV = VWMAG
      ISAV = 1
C
C  RESOLVE PERTURBATION VELOCITIES INTO WIND COORDINATES
C
      VWR = VSI*CPHI+WSI*SPHI
      WWR = -USI*SINAC-VSI*COSAC*SPHI+WSI*COSAC*CPHI
      UWR =  USI*COSAC-VSI*SINAC*SPHI+WSI*SINAC*CPHI+1.0
140   CONTINUE
      JXS = IXS+IM1
      XSI = A(JXS)
150   CONTINUE
      IF (ISAV.LT.NFLD) NFLD=MIN0(ISAV+5,NFLD)
C
C  COMPUTE INCREMENTAL CHANGE IN PRANDTL-MEYER TURNING ANGLE, DNU
C
      VW2 = VWR*VWR+WWR*WWR
      TANNU = SQRT(VW2)/UWR
      DNU= ATAN(TANNU)*DGRAD
      VW2 = VW2+UWR*UWR
C              TEST FOR EXPANSION IN FLOW
      IF (VW2.GT.1.0) DNU=-DNU
      ANGNU2 = ANGNU1-DNU
      SMALL = ANGNU2.LT.0.1
      IF (SMALL) GO TO 160
C
C  CALCULATE NEW MACH NUMBER AND BETA
C
      GO TO 170
C              IF SMALL, SET TO BSAVE VALUE
160   BETANU = BSAVE
      FNUMCH = FML
      GO TO 180
C              COMPUTE NEW MACH NUMBER
C              REFERENCE.  HALL, ROYAL AERONAUTICAL SOCIETY, SEPT 1975
170   YNU = (ANGNU2/130.450769)**(2./3.)
      YNUSQ = YNU*YNU
      YNUC  = YNU*YNUSQ
      FNUMCH =((1.+1.3604*YNU+0.0962*YNUSQ-0.5127*YNUC)
         /(1.-0.6722*YNU-0.3278*YNUSQ))*COS(DNU/DGRAD)
      SMALL = FNUMCH.LT.1.01
      IF (SMALL) GO TO 160
      BETANU = SQRT(FNUMCH*FNUMCH-1.)
      BETANU = TANAPB(BETANU,TANALP)
      IF (BETANU.GT.BTNOSE) GO TO 180
C              SET BETANU TO LIMIT AT NOSE IF GREATER THAN NOSE VALUE
      BETANU = BTNOSE
      FNUMCH = FML
C
C  COMPUTE SHOCK SHAPE BY INTEGRATION OF NEW MACH ANGLE
C
180   XSHK(ISHK) = XSHK(ISHK-1)+0.5*(BETANU+BSAVE)*(R-RSHK(ISHK-1))
      RSHK(ISHK) = R
      DXDR = (XSHK(ISHK)-XSHK(ISHK-1))/(R-RSHK(ISHK-1))
      NSHK(IRAD) = ISHK-MAXSHK*(IRAD-1)
      IF (IPGTO) WRITE(6,702) NSHK(IRAD),XSHK(ISHK),R,UWR,VWSAV,FNUMCH
     *   ,BETANU,DXDR,DNU,VW2
C
C  CONVERGENCE TESTS
C
      IF (XSHK(ISHK).GE.XMAX) GO TO 200
      IF (ABS(BETINF-BETANU)/BETINF .LE.0.01)  GO TO 200
      IF (.NOT.SMALL .AND. ABS(BETANU-BSAVE).LT.0.025) DR=DR+DR
      BSAVE = BETANU
      FML = FNUMCH
      IF (ISHK.GE.MAXSHK*IRAD) GO TO 200
      GO TO 100
200   CONTINUE
      RETURN
      END


      SUBROUTINE CONFIG (SFUS)
C
C     INPUT AND INITIALIZE CONFIGURATION DESCRIPTION
C
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSBY(51)
     *   ,FUSAZ(51),XJ(51),PHIK(33)
      COMMON /BOPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1   ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON /BSHOCK/ SHK(240)
C
C  GEOMETRY ARRAYS
      DIMENSION SFUS(1),CHAR(2)
      EQUIVALENCE (SHK(37),XSHLDR)
C
      LOGICAL LPRT,READA,READR,READE,READV,READH
      DATA CHAR(1),CHAR(2)/3HY= ,3HZ= /
      DATA PI/3.14159265/
      LPRT=IPRT(1).GT.0
C
C     REFERENCE AREA
C
      REFA=1.
      IF (J0.NE.0) READ (5,480) REFA
      IF (LPRT) WRITE(6,490) REFA
C
C     FUSELAGE (BODY)  ----------------------------------------------
C
      IF (J2.EQ.0) J2=-2
      J2TEST=3
C
C     J2 = 1, READ ARBITRARY Y,Z
C     J2 LT 0  INDICATE CIRCULAR/ELLIPTIC BODY WITH
C                J6 = 0,   CAMBERED BODY,  READ ZFUS
C                J6 = +-1, SYMMETRICAL BODY,  OMIT ZFUS
C     J2 = -1, READ BODY CROSS SECTIONAL AREAS VS XFUS ON INPUT
C     J2 = -2, READ BODY RADII VS XFUS ON INPUT
C     J2 = -3, READ ELLIPTIC BODY VERT AND HORIZ SEMI-MAJOR AXES VS XFUS
C     J2 = -4, READ ELLIPTIC BODY VERTICAL SEMI-AXIS AND RATIO VS XFUS
C     J2 = -5, READ ELLIPTIC BODY HORIZONTAL SEMI-AXIS AND RATIO VS XFUS
C
      IF (J2.LE.-1.AND.J6.EQ.-1) J2TEST=1
      IF (J2.LE.-1.AND.J6.EQ.0) J2TEST=2
      IF (J6.EQ.1) J2TEST=1
      READA=J2.EQ.-1
      READR=J2.EQ.-2
      READE=J2.EQ.-3
      READV = J2.EQ.-4
      READH = J2.EQ.-5
      J2=1
C
C     J2TEST = 1, CIRCULAR/ELLIPTIC FUSELAGE WITH NO BODY CAMBER
C     J2TEST = 2, CIRCULAR/ELLIPTIC FUSELAGE WITH BODY CAMBER
C     J2TEST = 3, ARBITRARY Y,Z CROSS-SECTION
C     READ XFUS
C
      I2 = 0
      L2 = 0
```

Figure A-1(i)

```
      DO 290 IFU=1,NFUS
      IF (LPRT) WRITE(6,570)
      NRAD=NRADX(IFU)
      NFUSOR=NFORX(IFU)
      I1 = I2+1
      I2 = I1+NFUSOR-1
      READ (5,480) (XFUS(I),I=I1,I2)
      IF (LPRT) WRITE(6,510) IFU,(XFUS(I),I=I1,I2)
      BODL = XFUS(I2)-XFUS(1)
C
C     J2TEST = 2 INDICATES CIRCULAR/ELLIPTIC CAMBERED FUSELAGE
C
      IF (J2TEST.NE.2) GO TO 190
      IF (LPRT) WRITE(6,580)
      READ (5,480) (ZFUS(I),I=I1,I2)
      IF (LPRT) WRITE(6,510) IFU,(ZFUS(I),I=I1,I2)
      GO TO 210
190   DO 200 I=I1,I2
      ZFUS(I) = 0.
200   CONTINUE
C
C     J2TEST = 3 INDICATES ARBITRARY FUSELAGE - READ YJ + ZJ OF SECTION
C
210   IF (J2TEST.NE.3) GO TO 250
      DO 230 JX=1,NFUSOR
      IF (LPRT) WRITE(6,590) JX
      DO 220 K=1,2
      L1 = L2+1
      L2 = L1+NRAD-1
C             K=1,  READ Y VALUES VS MERIDIAN ANGLE
C             K=2,  READ Z VALUES VS MERIDIAN ANGLE
      IF (K.EQ.1) LY=L1
      IF (K.EQ.2) LZ=L1
      READ (5,480) (SFUS(J),J=L1,L2)
      IF (LPRT) WRITE(6,600) CHAR(K),(SFUS(J),J=L1,L2)
220   CONTINUE
      CALL ASECTN (SFUS(LY),SFUS(LZ),NRAD,FUSARD(I1))
      I1 = I1+1
230   CONTINUE
      GO TO 280
C
C  CIRCULAR/ELLIPTIC FUSELAGE - READ CROSS SECTIONAL AREAS/RADII
C
250   CONTINUE
      IF (LPRT.AND.READA) WRITE(6,610)
      IF (LPRT.AND.READR) WRITE(6,615)
      IF (LPRT.AND.READE) WRITE(6,680)
      IF (LPRT.AND.READV) WRITE(6,690)
      IF (LPRT.AND.READH) WRITE(6,680)
      DO 255 I=I1,I2
255   FUSAZ(I) = 0.
      IF (READA) READ (5,480) (FUSARD(I),I=I1,I2)
      IF (READR) READ (5,480) (FUSBY(I),I=I1,I2)
      IF (READV) READ (5,480) (FUSAZ(I),I=I1,I2)
      IF (READE.OR.READH) READ (5,480) (FUSBY (I),I=I1,I2)
      IF (LPRT.AND.READA) WRITE(6,510) IFU,(FUSARD(I),I=I1,I2)
      IF (LPRT.AND.READR) WRITE(6,510) IFU,(FUSBY(I),I=I1,I2)
      IF (LPRT.AND.READV) WRITE(6,510) IFU,(FUSAZ (I),I=I1,I2)
      IF (LPRT.AND.(READE.OR.READH)) WRITE(6,510) IFU,(FUSBY(I),I=I1,I2)
C
      IF (.NOT.(READE.OR.READV.OR.READH)) GO TO 268
      IF (READE) WRITE(6,690)
      IF (READV) WRITE(6,710)
      IF (READH) WRITE(6,720)
      IF (READE.OR.READH) READ(5,480) (FUSAZ(I),I=I1,I2)
      IF (            READV) READ(5,480) (FUSBY(I),I=I1,I2)
      IF (LPRT.AND.(READE.OR.READH)) WRITE(6,510) IFU,(FUSAZ(I),I=I1,I2)
      IF (LPRT.AND.READV)            WRITE(6,510) IFU,(FUSBY(I),I=I1,I2)
268   CONTINUE
      RATM1 = 1.0
      RATIO = 1.0
      DO 270 I=I1,I2
      IF (READA) FUSBY(I)=SQRT(FUSARD(I)/PI)
      IF (READR) FUSARD(I)=PI*FUSBY(I)**2
      IF (READA.OR.READR) FUSAZ(I)=FUSBY(I)
      IF (READV) RATIO=FUSBY(I)
      IF (READH) RATIO=FUSAZ(I)
      IF (RATIO.LE.0.) RATIO=RATM1
      IF (READV) FUSBY(I)=FUSAZ(I)*RATIO
      IF (READH) FUSAZ(I)=FUSBY(I)*RATIO
      IF (READE.OR.READV.OR.READH) FUSARD(I)=PI*FUSAZ(I)*FUSBY(I)
      RATM1 = RATIO
270   CONTINUE
      IF (LPRT.AND.READV) WRITE(6,680)
      IF (LPRT.AND.READV) WRITE(6,510) IFU,(FUSBY(I),I=I1,I2)
      IF (LPRT.AND.READH) WRITE(6,690)
      IF (LPRT.AND.READH) WRITE(6,510) IFU,(FUSAZ(I),I=I1,I2)
280   CONTINUE
290   CONTINUE
C
C  FIND MAXIMUM CROSS SECTIONAL AREA AND SHOULDER AREA AND LOCATION
C
      ARMAX = 0.0
      DO 300 I=1,I2
300   ARMAX = AMAX1(ARMAX,FUSARD(I))
      ASHLD = 0.0
      IF (XSHLDR.GT.0.0) GO TO 330
      DO 310 I=2,I2
      IF (XFUS(I).GT.XFUS(I-1).AND.FUSARD(I).LE.ASHLD) GO TO 320
310   ASHLD = FUSARD(I)
      I = I2
320   XSHLDR = XFUS(I-1)
330   WRITE(6,730) ARMAX,ASHLD,XSHLDR,BODL
      RETURN
C
480   FORMAT (10F7.0)
490   FORMAT(///10X,26HVEHICLE GEOMETRY DEFINTION,74X,12H** CONFIG **
     *   ,/5X,33HREFERENCE AREA (J0.GT.0)     REFA=,G13.6)
510   FORMAT(I5,10F10.4,5(/5X,10F10.4))
570   FORMAT(//5H  NFU,5X,22HXFUS - BODY X-STATIONS)
580   FORMAT(/5H  NFU,5X,33HZFUS - BODY Z-CAMBER VS XFUS FOR
     *   ,16HCIRCULAR SECTION)
590   FORMAT(/5H  JX=,I3,5X,33HARBITRARY BODY COORDINATES (SFUS))
600   FORMAT(2X,A3,10F10.4,3(/5X,10F10.4))
610   FORMAT(/5X,41HFUSARD - CIRCULAR BODY CROSS SECTION AREA)
615   FORMAT(/5X,28HFUSRAD - CIRCULAR BODY RADII)
680   FORMAT(/5H  NFU,5X,43HFUSBY - ELLIPTIC BODY HORIZ SEMI-MAJOR AXES)
690   FORMAT(/5H  NFU,5X,42HFUSAZ - ELLIPTIC BODY VERT SEMI-MINOR AXES)
710   FORMAT(/5H  NFU,5X,32HELLIPTIC RATIO (HORIZ/VERT AXES))
720   FORMAT(/5H  NFU,5X,32HELLIPTIC RATIO (VERT/HORIZ AXES))
730   FORMAT(//10X,21HCROSS SECTIONAL AREAS
     *   /15H MAXIMUM AREA =,G12.5,5X,15HSHOULDER AREA =,G12.5,
     *   5X,12HX-SHOULDER =,G12.5,5X,13HBODY LENGTH =,G12.5)
      END


      SUBROUTINE DOUBLT(TX)
C
C SUBROUTINE TO CALCULATE THE STRENGTH OF A LINEAR LINE DOUBLET OF
C UNIT SLOPE WITH ORIGIN AT TX(J).
C
      COMMON /FLOW/DUM1(5),BETA,BETASQ,DUM2
      COMMON/SRCE/XFIELD,RFIELD,BSQ,X2,       U,V,VT
C
      X1=XFIELD-TX
      BR=BETA*RFIELD
      IF(X1.LE.BR    )GO TO 10
      XBR=X1/BR
      XX=SQRT(XBR*XBR-1.)
      U=BETA*XX
      ACOSH=ALOG(XBR+XX)
      XX=XBR*XX
      V=-.5*BETASQ*(ACOSH-XX)
      RETURN
C
```

Figure A-1(j)

```
C FIELD POINT IS AHEAD OF MACH CONE FROM DOUBLET ORIGIN.
C
   10 U=0.
      V=0.
      RETURN
      END


      SUBROUTINE DPCOEF
C
C     SUBROUTINE TO CALCULATE THE LEFT HAND SIDE OF THE EQUATIONS
C     FOR U*/VINF ON THE WING, PYLON AND FUSELAGE INTERFERENCE PANELS
C
      COMMON FVN(200,201)
      COMMON/BPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1    SANGW
      COMMON /CONFIG/NFU,NPY,NSTRS,LVSWP,NRACK
      COMMON/ICVEL/UP,VP,WP,II,IF,          DELTP(200)
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON/WGEOM/XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1     PSIWTE(20),Y(20),PHID(20),ZDIMED,WICR
      COMMON/WPGEOM/XRF(200),XRB(200),XLF(200),XLB(200),YRC(200),
     1  YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2     SNPHI(200),CSPHI(200)
C
C
C     I IS THE INDEX OF THE INFLUENCING PANEL
C
C
C     INFLUENCE OF WING PANELS
C
      DO 100 I=1,NPANLS
      II=I
C
C     WING PANELS ON WING CONTROL POINTS
C
      DO 50 J=1,NPANLS
      CALL VELWP1(XCPT(J),YCPT(J),ZCPT(J))
   50 FVN(J,I)=WP*CSPHI(J)-VP*SNPHI(J)
C
C     WING PANELS ON PYLON CONTROL POINTS
C
      IF(NPY.EQ.0) GO TO 61
      DO 60 J=N1P,N2
      CALL VELWP1(XCPT(J),YCPT(J),ZCPT(J))
   60 FVN(J,I)=-VP
   61 CONTINUE
C
C     WING PANELS ON FUSELAGE CONTROL POINTS
C
      IF(NFU.EQ.0) GO TO 100
      DO 80 J=N2P,NPTOT
      CALL VELWP1(XCPT(J),YCPT(J),ZCPT(J))
      NB=J-N2
      WW=-WP*CST2(NB)+VP*SNT2(NB)
   80 FVN(J,I)=WW
  100 CONTINUE
C
C     INFLUENCE OF PYLON PANELS
C
      IF(NPY.EQ.0) GO TO 201
      DO 200 I=N1P,N2
      II=I
C
C     PYLON PANELS ON WING CONTROL POINTS
C
      DO 150 J=1,NPANLS
      CALL VELPP1(XCPT(J),YCPT(J),ZCPT(J))
  150 FVN(J,I)=WP*CSPHI(J)-VP*SNPHI(J)
C
C     PYLON PANELS ON PYLON CONTROL POINTS
C
      DO 160 J=N1P,N2
      CALL VELPP1(XCPT(J),YCPT(J),ZCPT(J))
  160 FVN(J,I)=-VP
C
C     PYLON PANELS ON FUSELAGE CONTROL POINTS
C
      IF(NFU.EQ.0) GO TO 200
      DO 180 J=N2P,NPTOT
      CALL VELPP1(XCPT(J),YCPT(J),ZCPT(J))
      NB=J-N2
      WW=-WP*CST2(NB)+VP*SNT2(NB)
  180 FVN(J,I)=WW
  200 CONTINUE
C
C     INFLUENCE OF FUSELAGE INTERFERENCE PANELS
C
  201 IF(NFU.EQ.0) RETURN
      DO 300 I=N2P,NPTOT
      II=I
C
C     FUSELAGE PANELS ON WING CONTROL POINTS
C
      DO 250 J=1,NPANLS
      CALL VELBD1(XCPT(J),YCPT(J),ZCPT(J))
  250 FVN(J,I)=WP*CSPHI(J)-VP*SNPHI(J)
C
C     FUSELAGE PANELS ON PYLON CONTROL POINTS
C
      IF(NPY.EQ.0) GO TO 261
      DO 260 J=N1P,N2
      CALL VELBD1(XCPT(J),YCPT(J),ZCPT(J))
  260 FVN(J,I)=-VP
  261 CONTINUE
C
C     FUSELAGE PANELS ON FUSELAGE CONTROL POINTS
C
      DO 280 J=N2P,NPTOT
      CALL VELBD1(XCPT(J),YCPT(J),ZCPT(J))
      NB=J-N2
      WW=-WP*CST2(NB)+VP*SNT2(NB)
  280 FVN(J,I)=WW
  300 CONTINUE
      RETURN
      END


      SUBROUTINE DPRHS
C
C     SUBROUTINE TO CALCULATE THE RIGHT HAND SIDE OF THE EQUATIONS
C     FOR U*/VINF ON THE WING, PYLON, AND FUSELAGE INTERFERENCE PANELS
C
      LOGICAL CENTER,ZDIMED
C
      COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7),
     1 SSUMK(7),SSUMKD(7)
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /BPGEOM/ SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1      SANGW
      COMMON /CAMBER/ ALPHAL(200)
      COMMON /CONFIG/ NFU,NPY,NSTRS,LVSWP,NRACK
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITM,IDEL,NTAP7
     *    ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *    ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IA0,ID0,ISK0,IZ2(4),NRING,IROW(50)
      COMMON /EXVEL / UEI(200),VEI(200),WEI(200),CIR(200)
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ,FMCHSQ
      COMMON /FSOR  / FXL(101),FSS(100),FDS(100),NFSOR
```

Figure A-1(k)

```
      COMMON /ICVEL / UP,VP,WP,II,IF,        DELTP(200)
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBOCR1,NBOCR2,NBO,NBIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON /PYGEOM/ Z(20),XPLE,YPL,CRP,MP,PSIPLE(20),PSIPTE(20),IP,
     1      SLLE,PSLPOF,CENTER,ZPL,LVSPP
      COMMON/THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     1   XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2   THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3      YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4      PSPSTE(20),SNPHS(400),CSPHS(400)
      COMMON /WGEOM / XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1      PSIWTE(20),Y(20),PHID(20),ZDIMED,WICR
      COMMON /WPGEOM/ XRF(200),XRB(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2      SNPHI(200),CSPHI(200)
      COMMON            A(12000)
C
C     SET ALL EXTERNALLY INDUCED VELOCITIES EQUAL TO ZERO
C
      DO 90 J=1,NPTOT
      UEI(J)=0.0
      VEI(J)=0.0
   90 WEI(J)=0.0
C
C     FUSELAGE-INDUCED VELOCITIES ON WING AND PYLON CONTROL POINTS
C             USE LINE SOURCE METHOD
C
      IF (NFU.NE.1) GO TO 105
      DO 100 J=1,N2
      XVC=XCPT(J)+XBWOC
      ZVC=ZCPT(J)+ZBWO
      CALL VELCAL(FSS,FDS,FXL,NFSOR,XVC,YCPT(J),ZVC,UQ,VQ,WQ,
     1 FBASE,FSUMK,FSUMKD)
      UEI(J)=UEI(J)+UQ
      VEI(J)=VEI(J)+VQ
  100 WEI(J)=WEI(J)+WQ
      GO TO 110
C
C     FUSELAGE-INDUCED VELOCITIES - SOURCE PANELING METHOD
C
105   IF (NFU.NE.2) GO TO 110
C             DEFINE COORDINATES OF FIELD POINTS
C             POSITIVE VELOCITY SIGNS%  STORE COORDINATES - U-FORWARD
C                   V-RIGHT,  AND W-DOWN?  SOURCE PANELS - U-REAR
C                   V-RIGHT , AND W-UP.
C
C             RESTORE FUSELAGE ARRAYS IF NECESSARY
      REWIND 10
      IF (NSTRS.GT.0) CALL FRSTRT(10,2,LASTA)
C
      IXVC = IU
      IYVC = IXVC+N2
      IZVC = IYVC+N2
      ISVN = IZVC+N2
      ISKP = ISVN+N2
C
C     LOCATE POINTS IN NON-CIRCULAR FUSELAGE COORDINATE SYSTEM
C
      DO 106 J=1,N2
      XVC=XCPT(J)+XBWOC
      ZVC=ZCPT(J)+ZBWO
      JM1 = J-1
      JXVC = IXVC+JM1
      JZVC = IZVC+JM1
      A(JXVC) = -XVC
      A(JZVC) = -ZVC
106   CONTINUE
C
C  COMPUTE INFLUENCE OF BODY ON WING CONTROL POINTS FOR SOURCE PANELS.
C  NO BETA CORRECTION DUE TO NONLINEAR MACH CONE INCLUDED.
C
      CALL FLOVEL (A(IXVC),YCPT,A(IZVC),A(ISVN),A(ISKP)
     *    ,UEI,VEI,WEI,N2,NX,1)
C             TRANSFORM VELOCITIES BACK INTO STORE COORDINATES
C
      DO 108 J=1,N2
      UEI(J) = -UEI(J)
      WEI(J) = -WEI(J)
108   CONTINUE
C
C     THICKNESS INDUCED VELOCITIES
C       NOTES NO PYLON-ON-PYLON VELOCITIES CALCULATED.
C
C     WING ON WING CONTROL POINTS
C
110   CONTINUE
      IF(ZDIMED) GO TO 149
      II=1
      IF=MS
      DO 145 J=1,NPANLS
      CALL VELWT1(XCPT(J),YCPT(J),ZCPT(J))
      UEI(J)=UEI(J)+UP
      VEI(J)=VEI(J)+VP
      WEI(J)=WEI(J)+WP
  145 CONTINUE
C
C     IF PYLON NOT PRESENT, SKIP ALL PYLON CALCULATIONS
C
C
  149 IF(NPY .EQ. 0) GO TO 152
      II=MS+1
      IF=NTHP
C
C     PYLON ON WING CONTROL POINTS
C
C         J IS THE CONTROL POINT INDEX
C
      DO 150 J=1,NPANLS
      CALL VELPT1(XCPT(J),YCPT(J),ZCPT(J))
      UEI(J)=UEI(J)+UP
      VEI(J)=VEI(J)+VP
      WEI(J)=WEI(J)+WP
  150 CONTINUE
C
C     WING ON PYLON CONTROL POINTS
C
      II=1
      IF=MS
      DO 151 J=N1P,N2
      CALL VELWT1(XCPT(J),YCPT(J),ZCPT(J))
      UEI(J)=UEI(J)+UP
      VEI(J)=VEI(J)+VP
      WEI(J)=WEI(J)+WP
  151 CONTINUE
C
C     WING AND PYLON ON FUSELAGE CONTROL POINTS
C
  152 IF (NFU .EQ. 0) GO TO 111
C
C                 WING THICKNESS
C
      II=1
      IF=MS
      DO 154 J=N2P,NPTOT
      CALL VELWT1(XCPT(J),YCPT(J),ZCPT(J))
      UEI(J)=UEI(J)+UP
      VEI(J)=VEI(J)+VP
      WEI(J)=WEI(J)+WP
  154 CONTINUE
C
C                 PYLON THICKNESS
C
      IF (NPY .EQ. 0) GO TO 111
      II=MS+1
      IF=NTHP
      DO 155 J=N2P,NPTOT
      CALL VELPT1(XCPT(J),YCPT(J),ZCPT(J))
      UEI(J)=UEI(J)+UP
```

Figure A-1(1)

```
      VEI(J)=VEI(J)+VP
      WEI(J)=WEI(J)+WP
  155 CONTINUE
C
C     CALCULATE RIGHT HAND SIDE
C
C     -- WING BOUNDARY CONDITION --
C
  111 CONTINUE
      DO 120 J=1,NPANLS
  120 CIR(J)=(ALFACR+ALPHAL(J)+WICR)*CSPHI(J)+VEI(J)*SNPHI(J)-
     1       (UEI(J)*ALPHAL(J)+WEI(J))*CSPHI(J)
C
C     -- PYLON BOUNDARY CONDITION --
C
      IF(NPY.EQ.0) GO TO 131
      DO 130 J=N1P,N2
      IF(CENTER) VEI(J)=0.
  130 CIR(J)=VEI(J)
C
C     -- FUSELAGE INTERFERENCE PANEL BOUNDARY CONDITION --
C        ONLY V NORMAL IS USED IN FUSELAGE BOUNDARY CONDITION
C
  131 IF(NFU.EQ.0) RETURN
      DO 140 J=N2P,NPTOT
      NB=J-N2
      WW=-VEI(J)*CST2(NB)+VEI(J)*SNT2(NB)
  140 CIR(J)=-WW
      RETURN
      END


      SUBROUTINE FLDVEL (XFP,YFP,ZFP,SVN,LSKP,U,V,W,NFLD,ID,IG)
C
C  ROUTINE TO COMPUTE U,V,W VELOCITY COMPONENTS AT THE FIELD POINTS
C  XFP,YFP,ZFP.  ALL FIELD POINTS HAVE THE SAME  BETA EXCEPT THE
C  DESGNATED INLET PANELS.  THEY USE MINMUM OF BETA0 AND BTINLT
C
      COMMON           A(12000)
      LOGICAL LSKP(NFLD)
      DIMENSION XFP(NFLD),YFP(NFLD),ZFP(NFLD),U(NFLD),V(NFLD),W(NFLD)
     *   ,SVN(NFLD),ID(130)
      DATA IDLAST/-1/
C
C  INITIALIZE VELOCITY COMPONENTS TO ZERO.
C
      DO 100 I=1,NFLD
      LSKP(I) = .FALSE.
      U(I) = 0.
      V(I) = 0.
  100 W(I) = 0.
C           DEFINE INDEX OF FIRST LOCATION IN A-ARRAY OF PANEL
C           GEOMETRY AND STRENGTH
      IA0 = ID(73)
      ISK0 = ID(75)
      IXPT = ID(42)+IA0
      IYPT = ID(43)+IA0
      IZPT = ID(44)+IA0
      ITH  = ID(45)+IA0
      IDEL = ID(46)+IA0
      IGB  = ID(IG+50)+IA0
      KFUS = ID(12)
      IXZSYM = ID(38)
      IDLAST = IA0
C
C  SUM INFLUENCES OF EACH BODY SEGMENT ON FIELD POINTS
C
  120 JG = 0
      DO 150 IFU=1,KFUS
      JR = ID(IFU+12)-1
      IXC = ID(IFU+22)+IA0
      IYC = ID(IFU+27)+IA0
      IZC = ID(IFU+32)+IA0
C
C             LOOP THROUGH NUMBER OF RINGS IN SEGMENT OF INFLUENCING
C             PANEL BLOCKS.  L - INDEX OF CROSS SECTION IN THIS SEGMENT
C
      KFUSOR = ID(IFU+17)
      DO 140 L=2,KFUSOR
      CALL FLDVL2 (A(IXPT),A(IYPT),A(IZPT),A(ITH),A(IDEL),A(IGB)
     *   ,XFP,YFP,ZFP,U,V,W,SVN,LSKP,NFLD
     *   ,A(IXC),A(IYC),A(IZC),KFUSOR,IXZSYM,JR,JG,L)
  140 JG = JG+JR
  150 CONTINUE
      RETURN
      END


      SUBROUTINE FLDVL2 (XPT,YPT,ZPT,THET,DELTA,GB,XFP,YFP,ZFP,U,V,W
     *   ,SVN,LSKP,NFLD,XC,YC,ZC,KFUSOR,IXZSYM,JR,JG,L)
C
C  COMPUTE THE THREE COMPONENTS OF VELOCITY INDUCED AT SPECIFIED
C  CONTROL POINTS BY THE BODY PANELS OF A GIVEN BODY SEGMENT
C
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIV0,NINBLK,BTINLT,YCPI
     *   ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BLKPAN/ COST,SINT,XBTJ,YBTJ,ZBTJ,XC1,YC1,ZC1
     *   ,XPT1,YPT1,ZPT1,COSTI,SINTI,COSD,SIND,LZERO
      COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4)
     1   ,XI,YI,ZI,XJ,ZJ,BETA0,BETAL,SUBSON,SUPERS,J
      COMMON /BOPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
C
      DIMENSION XPT(20),YPT(20),ZPT(20),THET(20),DELTA(20)
     *   ,GB(20),U(20),V(20),W(20),XFP(20),YFP(20),ZFP(20),SVN(20)
     *   ,XC(KFUSOR),YC(KFUSOR,1),ZC(KFUSOR,1)
      LOGICAL LSKP(1)
      LOGICAL LZERO,SUBSON,SUPERS
      LOGICAL INLTST,INLET,NOINLT,OPEN
      EXTERNAL INLTST
      LZERO = .FALSE.
C             FOR FIELD VELOCITY CALCULATIONS, INCLUDE BOTH INLET AND
C             ADJACENT SET OF PANELS
      NTINL = NINLET+NINBLK+NINVEL
C
      SINTI = 0.
      COSTI = 1.
      SIND  = 0.
      COSD  = 1.
      XCOR(1)=0.
      YCOR(1)=0.
      ZCOR(1)=0.
      XCOR(3)=0.
C  JG IS STARTING OFFSET OF GB(J)
C  I IS THE INDEX OF THE FIELD POINT
C  J IS THE INDEX OF THE INFLUENCING PANEL
C
C  INITIALIZE SKIP LOGIC CHECK ON VELOCITY CONTRIBUTION
C  OF SUBSEQUENT PANELS OF A SINGLE RING
      DO 20 I=1,NFLD
   20 SVN(I) = 0.
C
C  SETUP LOCAL GEOMETRY FOR INFLUENCING PANEL OF RING JB
C
      DO 60 N=1,JR
      J = JG+N
      XBTJ = XPT(J)
      YBTJ = YPT(J)
      ZBTJ = ZPT(J)
      TAND=TAN(DELTA(J))
      COST=COS(THET(J))
      SINT=SIN(THET(J))
```

Figure A-1(m)

```
      GBJ = GB(J)
C
C  FOR INLET, SET TO GREATER OF BTINLT OR PANEL SLOPE
C
      BETAL = BETA0
      NOINLT = .NOT.INLTST(J,OPEN)
      IF (NOINLT) GO TO 30
      BETAL = AMIN1(BTINLT,BETA0)
      IF (BTINLT*TANO.GT.1.) BETAL=0.99/TANO
C
30    NP1 = N+1
      LM1 = L-1
      XC1=XC(LM1)
      YC1=YC(LM1,NP1)
      ZC1=ZC(LM1,NP1)
C
C  CALCULATION OF PANEL CORNER POINTS IN PANEL COORDINATE SYSTEM
C              K= 1         LM1= L-1        NP1= N+1
C                 2              L               N+1
C                 3              L-1             N
C                 4              L               N
C
      XCOR(2)=XC(L)-XC1
      XCOR(4)=XCOR(2)
      DO 40 K=2,4
      LM1 = L
      NP1 = N+1
      IF (K.GE.3) NP1=N
      IF (K.EQ.3) LM1=L-1
      DELY=YC(LM1,NP1)-YC1
      DELZ=ZC(LM1,NP1)-ZC1
      YCOR(K)=DELY*COST+DELZ*SINT
      ZCOR(K)=DELZ*COST-DELY*SINT
40    CONTINUE
      CX=XCOR(2)
C              LOOP COLUMNWISE THROUGH ALL FIELD POINTS
      DO 50 I=1,NFLO
      LZERO = .FALSE.
      IF (LSKP(I).AND.NOINLT) GO TO 50
      XPTI = XFP(I)
      YPTI = YFP(I)
      ZPTI = ZFP(I)
45    CALL PANVEL (UBI,VBI,WBI,ANI,IXZSYM)
      U(I) = U(I)+UBI*GBJ
      V(I) = V(I)+VBI*GBJ
      W(I) = W(I)+WBI*GBJ
      SVM(I) = SVM(I)+UBI*UBI+VBI*VBI+WBI*WBI
      IF (I.GT.1) LZERO=.TRUE.
50    CONTINUE
60    CONTINUE
C
C  CHECK WHETHER CONTRIBUTION OF RING IS ZERO.
      IF (SUBSON) GO TO 110
      DO 70 I=1,NFLO
70    LSKP(I) = SVM(I).EQ.0.
110   CONTINUE
      RETURN
      END


      SUBROUTINE FRSTRT (IO,KODE,LASTA)
C
C  ROUTINE TO SAVE OR RESTORE REQUIRED PROGRAM INFORMATION FOR SOURCE
C  PANELING METHOD TO RESTART CONFIGURATION ANALYSIS.
C        IO    = FILE UNIT NUMBER FOR DATA STORAGE.
C          KODE = 1, SAVE CONTROL INTEGERS AND VARIABLES
C          KODE = 2, RESTORE CONTROL INTEGERS AND VARIABLES
C          KODE = 3, SAVE ABOVE AND AIC,U,V,W MATRICES
C          KODE = 4, RESTORE ABOVE AND AIC,U,V,W MATRICES
C          KODE = 5, RESTORE CONTROL INTEGERS AND VARIABLES INTO
C                    CORE AFTER EXISTING DATA IN BLANK COMMON
C          KODE = 6, RESTORE ABOVE AND SKIP PAST TAPE8 + TAPE9 DATA
C  MULTIPLE CASES MAY BE STACKED BACK TO BACK.  THE CALLING PROGRAM IS
C  RESPONSIBLE FOR PROPER POSITIONING OF SEQUENTIAL CASES.
C
      COMMON /BGEOM / GEOMB(339)
      COMMON /BOPTNS/ IOPTS(50)
      COMMON /BSHOCK/ SHOCKB(240)
      COMMON /DIMENS/ IDIM(79),NRING,IROW(50)
      COMMON /HEAD  / TITLE(40)
      COMMON /PARAM / BPARAM(12)
      COMMON /BINLET/ AINLET(38)
      COMMON /BINSHK/ AINSHK(189)
      COMMON              A(12000)
      DIMENSION IA(130)
      EQUIVALENCE (NTAP7,IDIM(47)),  (INLET,IOPTS(18))
      EQUIVALENCE (IA(1),A(1),LA(1)),  (IUVW,IOPTS(28))
      LOGICAL INLET,SKIP,LA(1),INLETA
C
      GO TO (100,200,100,200,300,300),KODE
C
C  SAVE COMMON BLOCK AND TAPE7 ON TAPE-IO
100   CONTINUE
      CALL IOWRIT (IO,IDIM,130)
      CALL IOWRIT (IO,BPARAM,12)
      CALL IOWRIT (IO,IOPTS,50)
      CALL IOWRIT (IO,GEOMB,339)
      CALL IOWRIT (IO,TITLE,40)
      CALL IOWRIT (IO,SHOCKB,240)
      IF (INLET) CALL IOWRIT (IO,AINLET,38)
      IF (INLET) CALL IOWRIT (IO,AINSHK,189)
      CALL IOWRIT (IO,A,NTAP7)
      IF (KODE.NE.3)                    RETURN
C
C  COPY AN AND U,V,W  MATRICES FROM TAPE9 AND TAPE8
      REWIND 8
      REWIND 9
      CALL IOREAD (9,IROW,NRING)
      SKIP = IUVW.NE.0
      DO 150 JB=1,NRING
      JR = IROW(JB)
      DO 150 IB=JB,NRING
      IR = IROW(IB)
      NA = IR*JR
C             AN - MATRIX
      CALL IOREAD ( 9,A,NA)
      CALL IOWRIT (IO,A,NA)
C             U,V,W
      IF (SKIP) GO TO 150
      CALL IOREAD ( 8,A,3*NA)
      CALL IOWRIT (IO,A,3*NA)
150   CONTINUE
      RETURN
C
C  RESTORE DATA FROM TAPE - IO  ----------------------------------------
C
200   CONTINUE
      CALL IOREAD (IO,IDIM,130)
      CALL IOREAD (IO,BPARAM,12)
      CALL IOREAD (IO,IOPTS,50)
      CALL IOREAD (IO,GEOMB,339)
      CALL IOREAD (IO,TITLE,40)
      CALL IOREAD (IO,SHOCKB,240)
      IF (INLET) CALL IOREAD (IO,AINLET,38)
      IF (INLET) CALL IOREAD (IO,AINSHK,189)
      CALL IOREAD (IO,A,NTAP7)
      REWIND 7
      CALL IOWRIT ( 7,A,NTAP7)
      LASTA = NTAP7
      IF (KODE.NE.4)                    RETURN
C
C  COPY AN AND U,V,W  MATRICES ONTO TAPE9 AND TAPE8
C
      REWIND 8
      REWIND 9
      CALL IOWRIT (9,IROW,NRING)
```

Figure A-1(n)

```
      SKIP = IUVW.NE.0
      DO 250 JB=1,NRING
      JR = IROW(JB)
      DO 250 IB=JB,NRING
      IR = IROW(IB)
      NA = IR*JR
C              AN - MATRIX
      CALL IOREAD (IO,A(LASTA+1),NA)
      CALL IOWRIT ( 9,A(LASTA+1),NA)
C              U,V,W
      IF (SKIP) GO TO 250
      CALL IOREAD (IO,A(LASTA+1),3*NA)
      CALL IOWRIT ( 8,A(LASTA+1),3*NA)
250   CONTINUE
      RETURN
C
C  RESTORE DATA FROM TAPE-IO TO BLANK COMMON ------------------------------
C
300   CONTINUE
      ID0 = LASTA
C                        IDIM
      CALL IOREAD (IO,IA(ID0+1),130)
      IA(ID0+74) = ID0
C                        NRING
      IF (KODE.EQ.6) NR=IA(ID0+80)
      IP0 = ID0+130
C                        BPARAM
      CALL IOREAD (IO,A(IP0+1),12)
      ISK0 = IP0+12
C                        IOPTS
      CALL IOREAD (IO,A(ISK0+1),50)
      INLETA = LA(ISK0+18)
      SKIP = IA(ISK0+28).NE.0
      ISK0 = ISK0+50
C                        GEOMB
      CALL IOREAD (IO,A(ISK0+1),339)
      ISK0 = ISK0+339
C                        TITLE
      CALL IOREAD (IO,A(ISK0+1),40)
      ISK0 = ISK0+40
C                        SHOCKB
      CALL IOREAD (IO,A(ISK0+1),240)
      IA(ID0+75) = ISK0
      IA0 = ISK0+240
      IA(ID0+73) = IA0
C                        INLET
      IF (INLETA) CALL IOREAD (IO,AINLET,38)
      IF (INLETA) CALL IOREAD (IO,AINSHK,189)
C                        A          NTAP7
      CALL IOREAD (IO,A(IA0+1),IA(ID0+47))
      LASTA = LASTA+IA0+IA(ID0+47)
C
C  FOR KODE=6, READ PAST U,V,W INFLUENCE MATRICES.
      IF (KODE.EQ.5)                       RETURN
      DO 350 JB=1,NR
      JR = ID0+JB+80
      DO 350 IB=JB,NR
      IR = ID0+IB+80
      NA = IA(JR)*IA(IR)
      CALL IOREAD (IO,A(LASTA+1),NA)
      IF (SKIP) GO TO 350
      CALL IOREAD (IO,A(LASTA+1),3*NA)
350   CONTINUE
      RETURN
      END


      SUBROUTINE FUSEIO
C
C
C     SUBROUTINE TO INPUT AND OUTPUT FUSELAGE DATA
C        LINE SOURCE AND DOUBLET DISTRIBUTIONS ARE CALCULATED
C        USING SUBROUTINE BOYGEN
C
C
      COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7),
     1 SSUMK(7),SSUMKD(7)
      COMMON /FLOW/ALFACH,GAMF,FMACH,RHO,VINF,BETA,BETASQ,FMCHSQ
      COMMON /FSGEOM/FRMAX,SSPAN,    FLTHC,BODYPL
      COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101)
      COMMON /FSOR/FXL(101),FSS(100),FDS(100),NFSOR
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NB1P,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON /SHPDAT/NFPOLY,FXEND(7),FCOEF(7,7)
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
701 FORMAT(10I5)
706 FORMAT(8F10.0)
708 FORMAT(10X,19HFUSELAGE INPUT DATA)
709 FORMAT(15X,17HFUSELAGE LENGTH =,F10.5,5H FEET/15X,16HMAXIMUM RADIU
   1S =,F10.5,5H FEET/15X,22HNOSE TIP SHOCK ANGLE =,F10.5,8H DEGREES/)
729 FORMAT(   15X,37HPOLYNOMIALS SPECIFYING FUSELAGE SHAPE
   1 /18X,26HX/L OF END OF EACH SECTION/21X,7HSECTION,5X,3HX/L)
730 FORMAT(23X,I2,3X,7F10.5)
731 FORMAT(18X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTION/
   121X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X,
   22HC7)
732 FORMAT(/14X,I3,  85H RINGS OF U-VELOCITY PANELS ARE TO BE LAID OUT
   1 ON THE FUSELAGE FROM X= 0 FEET TO X=,F10.5,5H FEET/20X,9HTHERE
   2ARE,I3,54H PANELS IN EACH RING ON THE LEFT HALF OF THE FUSELAGE,,
   3I3,19H ABOVE THE WING AND,I3,15H BELOW THE WING/20X,25HTOTAL NUMBE
   4R OF PANELS IS,I4)
733 FORMAT(/15X,105HFUSELAGE SHAPE AS CALCULATED FROM THE INPUT POLYNO
   1MIALS, ORIGINS OF SOURCES AND DOUBLETS REPRESENTING THE/20X,56HFUS
   2ELAGE, AND VALUES OF THE SOURCE AND DOUBLET CONSTANTS)
734 FORMAT(/15X,25HFUSELAGE SHOCK WAVE SHAPE)
C
C     READ IN AND WRITE FUSELAGE GEOMETRY AND POLYNOMIALS
C        DESCRIBING SHAPE
C
      READ (5,706) FLTHC,FRMAX,FTHSHK
      WRITE (6,708)
      WRITE (6,709) FLTHC,FRMAX,FTHSHK
      READ (5,701) NFPOLY
      READ (5,706) (FXEND(J),J=1,NFPOLY)
      DO 3 J=1,NFPOLY
    3 READ (5,706) (FCOEF(J,K),K=1,7)
      WRITE (6,729)
      DO 4 J=1,NFPOLY
    4 WRITE (6,730) J,FXEND(J)
      WRITE (6,731)
      DO 5 J=1,NFPOLY
    5 WRITE (6,730) J,(FCOEF(J,K),K=1,7)
C
C     READ AND WRITE FUSELAGE PANEL DATA
C
      READ (5,701) NCWB,NBDCR1,NBDCR2,NFSOR
      READ (5,706) BODYPL
      NBD=NBDCR1+NBDCR2
      NFTOT=NCWB*NBD
      BODYPL=ABS(BODYPL)
      BB=-BODYPL
      BTNOSE=TAN((90.0-FTHSHK)/57.2957795)
      WRITE (6,732) NCWB,BB,NBD,NBDCR1,NBDCR2,NFTOT
C
C     CALCULATE LINE SOURCE AND DOUBLET DISTRIBUTIONS
C
      NXBODY=NFSOR+1
      WRITE (6,733)
      CALL BOYGEN(NXBODY,FRMAX,FLTHC,NFPOLY,FXEND,FCOEF,FSS,FDS,FXL,
     1 ALFACR,FDRDX,FBASE,FSUMK,FSUMKD)
      NFSOR=NXBODY-1
C
C     FIND FUSELAGE SHOULDER LOCATION
C
      DO 10 N=1,NFSOR
      NN=N
```

Figure A-1(o)

```
      IF (FORDX(N).LE.0.0) GO TO 11
   10 CONTINUE
   11 FXSHLD=FXL(NN)
C
C     CALCULATE FUSELAGE SHOCK WAVE SHAPE
C
      WRITE (6,734)
      CALL SHKSHP(NFSOR,FXL,FSS,BETA,FRMAX,BTNOSE,NFSHK,FXSHK,FRSHK)
      RETURN
      END



      SUBROUTINE GEOM
C
C   INPUT CONFIGURATION GEOMETRY AND COMPUTE PANELS
C
C  TAPE USAGE IN GEOM-
C    TAPE5%  GEOM,     READ INPUT
C
C    TAPE7%  GEOM,     REWIND
C            BODPAN,   WRITE  XPT,.ZC    - SAVE BODY GEOMETRY
C
C    TAPE8%  GEOM,     REWIND
C            NEWRAD,   WRITE YB,ZB      - PER SEGMENT
C            BODPAN,   REWIND
C                      READ  YB,ZB      - PER SEGMENT
C                      REWIND
C
C  BLANK COMMON USAGE%
C
C  (1) AFTER CALL TO CONFIG
C       SFUS....
C  (2) AFTER CALL TO NEWRAD
C       SFUS,YB,ZB....
C  (3) AFTER CALL TO BODPAN
C       XPT,YPT,ZPT,THET,DELTA,AREA,XC,YC,ZC,YB,ZB,....
C
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSRAD(51)
     *    ,FUSAZ(51),XJ(51),PHIK(33)
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI
     *    ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BINSHK/ NIS,XISHLO,EALPI,XCLOSD,MAXSHI,NINL(8),PHINL(8)
         ,YINL(8),XINL(80),RINL(80)
      COMMON /BIPYZ / XBIP,NYZBIP,YCB(33),ZCB(33)
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,DBETA
     *    ,EALPHA,CNU0,CNU2,XSHLDR,SHK(3),XSHK(100),RSHK(100)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAPT
     *    ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IY8
     *    ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IA0,ID0,ISK0,IZ2(4),NRING,IROW(50)
      COMMON /HEAD  / TITLE1(20),TITLE2(20)
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1    ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON            A(12000)
C
      DIMENSION ID(130)
      EQUIVALENCE (ID(1),NX)
      LOGICAL LPRT,NEWPHI,TEST,INLET
C
      NEWPHI = .FALSE.
      DO 20 I=1,32
   20 NRADX(I) = 0
      DO 21 I=1,130
   21 ID(I) = 0
      NADIM = 40200
      NXDIM = 51
      EM = -1.0
      NBODY=0
      IA0 = 0
      ID0 = 0
      ISK0 = 0
      ITMAX=20
      CCTEST=0.0001
C
      ALPHA = 0.0
      BETA  = 0.0
      REFAR=0.
      REFD=1.
      REFL=0.
      REFX=0.
      REFZ=0.
      XSHLDR = 0.0
      REWIND 7
      REWIND 8
C
C  INPUT CONFIGURATION PARAMETERS
C
      READ (5,150) TITLE1
   40 WRITE (6,170) TITLE1
C
C PRINT AND CONTROL OPTIONS
      READ (5,180) IXZSYM,IPRT,IUVW,NSHOCK,MAXSHK,NINLET,NINBLK,NINVEL
      LPRT = IPRT(1).GT.0
      NTINL = NINLET+NINBLK+NINVEL
      INLET = NTINL.NE.0
      IF (MAXSHK.EQ.0) MAXSHK=7
      IF (LPRT) WRITE(6,220) IXZSYM,IPRT,IUVW,NSHOCK,MAXSHK
     *    ,NINLET,NINBLK,NINVEL
C
C AXIAL CALCULATION PARAMETERS
      READ (5,160) XBIP,XSHLDR,EALPHA
      IF (LPRT) WRITE(6,230) XBIP,XSHLDR,EALPHA
C
C GEOMETRY DEFINTION OPTIONS
      READ (5,180) J0,J2,J6,NFUS,(NRADX(I),NFORX(I),I=1,NFUS)
      IF (J2.EQ.0) J2=-2
      IF (LPRT) WRITE(6,240) J0,J2,J6,NFUS,NRADX,NFORX
C
C COMPUTE ARRAY LENGTH
C
      NX = 0
      NXNR = 0
      MAXNX = 0
C
      DO 50 NFU=1,NF
      NX = NX+NFORX    U)
      NXNR = NXNR+'   RX(NFU)*NRADX(NFU)
      MAXNX = MAX(   AXNX,NFORX(NFU))
   50 CONTINUE
      TEST = NX.G   50
      IF (TEST) W-ITE(6,320)
      IF (TEST) STOP ? )
C                SEE VAR   BLES%  /BGEOM/  XFUS,ZFUS,FUSARD,FUSRAD,FUSAZ,XJ
C
C  READ EXTERNAL CON   URATION GEOMETRY
C
      CALL CONFIG (.
      READ (5,150)   TLE2
      IF (LPRT) WR   E(6,170) TITLE2
C
C   INPUT REVISED  ONFIGURATION PANELING DESCRIPTION CONTROL INTEGERS
C
      KFUS = NFUS
      READ (5,180) K0,(KRADX(I),KFORX(I),I=1,KFUS)
      IF (LPRT) WRITE(6,250) K0,KFUS,KRADX,KFORX
C
C
C  READ INLET PARAMETERS
C
      IF (INLET) READ(5,180) NIS,(JINLT(I),I=1,NTINL)
      IF (INLET.AND.LPRT) WRITE(6,380) NTINL,NIS
```

Figure A-1(p)

```
      IF (LPRT.AND.NINLET.GT.0) WRITE(6,390) (JINLT(I),I=1,NINLET)
      INP1 = NINLET+1
      INP2 = NINLET+NINBLK
      IF (LPRT.AND.NINBLK.GT.0) WRITE(6,400) (JINLT(I),I=INP1,INP2)
      INP1 = INP1+NINBLK
      IF (LPRT.AND.NINVEL.GT.0) WRITE(6,410) (JINLT(I),I=INP1,NTINL)
      IF (INLET) READ (5,160) BTINLT,RVIVO
      IF (INLET.AND.LPRT) WRITE(6,370) BTINLT
C
      IF (KO.EQ.0) GO TO 60
      READ (5,160) REFAR,REFD,REFL,REFX,REFZ
      IF (LPRT) WRITE (6,270) REFAR,REFD,REFL,REFX,REFZ
60    CONTINUE
      IF (REFAR.NE.0.) REFA=REFAR
      IF (REFA.LE.0.) REFA=1.0
      IF (REFD.EQ.0.) REFD=1.
      IF (REFL.EQ.0.) REFL=BODL
      KX = 0
      NXKR = 0
      NBODY = 0
      KXKR = 0
      MAXKR = 0
      NATOT = 0
      DO 65 NFU=1,NFUS
      KRAD = IABS(KRADX(NFU))
      IF (KRAD.EQ.0) KRAD=NRADX(NFU)
      IF (IXZSYM.GT.0) KRAD=2*KRAD-1
      KFUSOR = IABS(KFORX(NFU))
      IF (KFUSOR.EQ.0) KFUSOR=NFORX(NFU)
      KX = KX+KFUSOR
      KXKR = KXKR+KFUSOR*KRAD
      NXKR = NXKR+NFORX(NFU)*KRAD
      NBODY = NBODY+(KRAD-1)*(KFUSOR-1)
      MAXKR = MAX0(MAXKR,KRAD)
65    CONTINUE
      NG = 6*NBODY+KX+2*KXKR
      NAG = MAX0(NG,2*NXNR)+`*NXKR
      NATOT = NAG
C
C  CHECK ALLOWABLE DIMENSIONS
C
      TEST = MAXKR.GT.33
      IF (TEST) WRITE(6,330)
      IF (TEST) STOP 330
C             SEE VARIABLES%  /BGEOM/ - PHIK
C             AND NEWRAD - PHIK,ANCOS,ANSIN,YF,ZF
      TEST = NFUS.GT.5
      IF (TEST) WRITE(6,340)
      IF (TEST) STOP 340
C             SEE VARIABLES%  /BOPTNS/ NRADX,NFORX,KRADX,KFORX,
C             AND /DIMENS/ IXC,IYC,IZC
C
      IF (NATOT.GT.NADIM) WRITE(6,350) NATOT,NADIM
C
C  DEFINE MERIDIONAL ANGLES FOR SYMMETRIC HALF OF BODY  ----------------
C
      I2 = 0
      DO 100 IFU=1,NFUS
      KRAD = IABS(KRADX(IFU))
      IF (J2TEST.EQ.3 .AND. KRAD.EQ.0)  GO TO 100
      IF (KRAD.EQ.0) KRAD=NRADX(IFU)
      I1 = 1
      I2 = KRAD
      NEWPHI = KRADX(IFU).LT.0
      IF (NEWPHI)  GO TO 80
C             USE EVEN INCREMENT PHI
      FANG = FLOAT(KRAD-1)
      DELE = 180./FANG
      DO 70 K=I1,I2
70    PHIK(K) = DELE*FLOAT(K-1)
      GO TO 90
C
C             READ NEW MERIDIAN ANGLES FOR SEGMENT
80    READ (5,160) (F IK(K),K=I1,I2)
90    IF (LPRT) WRITE(6,300) IFU,(PHIK(K),K=I1,I2)
C
C   REVISE BODY (FUSELAGE) MERIDIAN LINE SPACING, THEN
C
100   CALL NEWRAD(IFU)
C
C  DEFINE NEW AXIAL STATIONS  ------------------------------------------
       I2 = 0
       IX2 = 0
       DO 130 IFU=1,NFUS
       IX1 = IX2+1
       IX2 = IX2+NFORX(IFU)
       KFUSOR = IABS(KFORX(IFU))
       IF (KFUSOR.EQ.0) KFUSOR=NFORX(IFU)
       I1 = I2+1
       I2 = I1+KFUSOR-1
       IF (KFORX(IFU)) 120,110,105
C
C   READ IN NEW AXIAL STATIONS (XJ) FOR BODY (FUSELAGE)
C
105    READ (5,160) (XJ(K),K=I1,I2)
       GO T  125
C
C  USE ORIGINAL AXIAL STATIONS XJ=XFUS
110    KFORX(IFU)=KFUSOR
       DO 115 K=I1,I2
       XJ(K)=XFUS(IX1)
115    IX1 = IX1+1
       GO TO 125
C
C  DEFINE EVEN INCREMENT AXIAL DIVISIONS OF SEGMENT
C
120    DX = (XFUS(IX2)-XFUS(IX1))/FLOAT(KFUSOR-1)
       DO 121 K=I1,I2
       KFORX(IFU) = KFUSOR
121    XJ(K) = XFUS(IX1)+DX*(K-I1)
125    IF (LPRT) WRITE(6,310) IFU,(XJ(K),K=I1,I2)
130    CONTINUE
C
C   REVISE AXIAL PANEL SPACING ON BODY (FUSELAGE) AND
C   COMPUTE NEW PANEL GEOMETRY
       NTINL = NINLET+NINBLK
       CALL BODPAN
C
       RETURN
C
C
150    FORMAT (20A4)
160    FORMAT (10F7.0)
170    FORMAT (1H1,//5X,20A4,25X,12H** GEOM   **)
180    FORMAT (14I5)
220    FORMAT(/
      * /5X,50HPANEL SYMMETRY OPTION                        (IXZSYM) = ,I5,
      * /55X,27HINPUT GEOM SHOCK VEL SOLN  ,
      * /5X,50HPRINT CONTROLS (NO=0,YES=*)                    (IPRT) = ,5I5,
      * /5X,50HU,V,W CALCULATION (YES=0,NO=1)                 (IUVW) = ,I5,
      * /5X,50HNUMBER OF SHOCK SHAPES                       (NSHOCK) = ,I5,
      * /5X,50HMAXIMUM NUMBER OF POINTS IN SHAPE            (MAXSHK) = ,I5,
      * /5X,50HOPEN INLET PANELS (0=NO, 1-25 PANELS)        (NINLET) = ,I5,
      * /5X,50HCLOSED INLET PANELS                          (NINBLK) = ,I5,
      * /5X,50HINLET PANELS IN FIELD VELOCITY CALC          (NINVEL) = ,I5)
230    FORMAT(/
      * /5X,50HX-LOC FOR INTERFERENCE SHELL GEOMETRY          (XBIP) =,G12.5,
      * /5X,50HNOSE SHOULDER LOCATION FOR SHOCK             (XSHLDR) =,G12.5,
      * /5X,50HRATIO FOR SHOCK ROTATION WITH ALPHA          (EALPHA) =,G12.5)
240    FORMAT(//15X,35HEXTERNAL GEOMETRY OPTIONS AND SHAPE
      * /5X,50HREAD REF. AREA CARD (NO=0,YES=1)                 (J0) = ,I5,
      * /5X,50HEXTERNAL GEOMETRY TYPE                           (J2) = ,I5,
      * /5X,50HREAD Z-CENTERLINE (NO=*-1, YES=0)                (J6) = ,I5,
      * /5X,50HNUMBER OF BODY SEGMENTS                        (NFUS) = ,I5,
      * /46X,9HSEGMENT  ,25H  (1)   (2)   (3)   (4)   (5) ,
      * /5X,50HNUMBER OF GEOMETRY CORNER MERIDIANS           (NRADX) = ,5I5,
      * /5X,50HNUMBER OF CROSS SECTION AXIALLY               (NFORX) = ,5I5)
250    FORMAT(//15X,43HREVISED PANELING GEOMETRY OPTIONS AND SHAPE
```

Figure A-1(q)

```
     * /5X,50HREAD REFERENCE LENGTH CARD (NO=0, YES=1)     (KO) = ,I5,
     * /5X,50HNUMBER OF BODY PANELING SEGMENTS           (KFUS) = ,I5,
     * /46X,9HSEGMENT  ,25H  (1)  (2)  (3)  (4)  (5) ,
     * /5X,50HNUMBER OF PANEL CORNERS RADIALLY          (KRADX) = ,5I5,
     * /5X,50HNUMBER OF PANEL CROSS SECTIONS AXIALLY    (KFORX) = ,5I5)
 270  FORMAT(//10X,23HPANEL REFERENCE LENGTHS
     *   ,/4X,5HREFAR,6X,4HREFD,6X,4HREFL
     *   ,6X,4HREFX,6X,4HREFZ,/8F10.3)
 300  FORMAT(/7H    IFU=,I5,/7H   PHIK=,10G12.5,2(/7X,10G12.5))
 310  FORMAT(/7H    IFU=,I5,/7H     AJ=,10G12.5,2(/7X,10G12.5))
 320  FORMAT(/5X,42HERROR - TOTAL NUMBER OF EXTERNAL GEOMETRY
     *   ,33H AXIAL STATIONS EXCEEDS 50 - GEOM)
 330  FORMAT(/5X,41HERROR - MAXIMUM NUMBER OF MERIDIANAL CUTS
     *   ,18H EXCEEDS 33 - GEOM)
 340  FORMAT(/5X,51HERROR - MAXIMUM NUMBER OF SEGMENTS EXCEEDS 5 - GEOM)
 350  FORMAT(/10X,44HWARNING - BLANK COMMON REQUIRED FOR GEOM    =,I6
     *   ,21H  EXCEEDS DIMENSION =,I6)
 370  FORMAT (
     * /5X,50HINLET SHOCK PARAMETER - BETA              (BTINLT) =,F10.5)
 380  FORMAT(
     * /5X,50HTOTAL NUMBER OF INLET PANELS               (NTINL) = ,I5,
     * /5X,50HNUMBER OF INLET SHOCK SHAPE TRAVERSES        (NIS) = ,I5)
 390  FORMAT(
     *   5X,50HOPEN INLET PANEL NUMBERS                  (JINLT) = ,
     *   5I5,3(/55X,5I5))
 400  FORMAT(
     *   5X,50HCLOSED INLET PANEL NUMBERS                (JINLT) = ,
     *   5I5,3(/55X,5I5))
 410  FORMAT(5X,50HPANEL NUMBERS IN FIELD VELOCITY CALC       (JINLT) = ,
     *   5I5,3(/55X,5I5))
      END


      SUBROUTINE INLSHK(PHISHK)
C
C  THIS SUBROUTINE CALCULATES THE SHAPE OF THE SHOCK WAVE PRODUCED
C  BY AN ARBITRARY BODY AT ANGLE OF ATTACK, ALPHAC, ALONG A GIVEN
C  MERIDIAN, THETA.
C  INPUTS
C    PHISHK = NSHOCK=0, ANGLE AT WHICH SHOCK TRAVERSE IS MADE
C             .GT.0,MAX MERIDIANL ANGLE TO WHICH SHOCK IS CALCULATED
C             ON EVEN SPACING
C
C  BLANK COMMON USAGE
C  (1) INITIALIZE GEOMETRY AND SOURCE STRENGTHS FROM TAPE7
C      XPT,YPT,ZPT,THET,DELTA,AREA,XC,YC,ZC,GB,U,V,W,VA,WA,CP,...
C  (2) FOR FIELD VELOCITY CALCULATION
C      XPT,....GB,XS,YS,ZS,US,VS,WS,SVN,LSKP,...
C
      LOGICAL SMALL,SUBSON,SUPERS,IPGTO
      LOGICAL INLTST,INLET,OPEN
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI
     *   ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BINSHK/ NIS,XISHLD,EALPI,XCLOSD,MAXSHI,NINL(8),PHINL(8)
     *   ,YINL(8),XINL(80),RINL(80)
      COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4)
     1   ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,J
      COMMON /BOPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,OBETA
     *   ,EALPHA,CNUU,CNU2,XSHLOP,SHK(3),XSHK(100),RSHK(100)
      COMMON /CONSTS/ PI,PI2,DTOR,RTOD,FOURPI
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *   ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAV,NAS,NASHK,NAFLO,IAO,IDO,ISKO,IZ2(4),NRING,IROW(50)
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1   ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON          A(12000)
C
C
C  DOUBLE ANGLE TRIG IDENTITY FOR TAN(A+B)
C
      TANAPB(TANA,TANB) = (TANA+TANB)/(1.0-TANA*TANB)
C
 701  FORMAT (///30X,47HMODIFIED SHOCK LOCATIONS RADIAL ANGLE (PHINL) =,
     *   F8.3,/30X,20HCENTROID AT$ XINLT =,F7.3,4X,6HYINL =,F7.3,
     *   4X,7HZINLT =,F7.3,
     *   //14X,1HN,5X,1HX,9X,1HR,9X,1HU,9X,1HW,9X,2HML,7X,4HBETA
     *   ,4X,5HDX/DP,7X,3HDNU,5X,5HUVWS2)
 702  FORMAT(10X,I5,2F10.3,9F10.5)
 708  FORMAT(///15X,26HFUSELAGE INLET SHOCK SHAPE,69X,12H** INLSHK **)
 709  FORMAT(//25X,14HINLET GEOMETRY,
     *   /15X,15HLEADING EDGE   ,9H  XINLT =,F8.3,10H    YINLT =,F8.3,
     *   10H    ZINLT =,F8.3,/15X,15HTRAILING EDGE  ,9H XINLTE =,F8.3,
     *   10H  YINLTE =,F8.3,10H  ZINLTE =,F8.3,
     *   /15X,24HMASS FLOW RATIO (RVIVO=),F8.3,10H  XCLOSD =,F8.3)
C
      EALPI = EALPHA
      MAXSHI = MAXSHK
      WDGSHK = 85.0
      IF (.NOT.INLET) RETURN
      IPGTO = IPRT(3).GT.0
      IF (IPGTO) WRITE (6,708)
C
C  INITIALIZATION  -----------------------------------------------------
C
      BETAO = SQRT(XMACH*XMACH-1.)
      BETAL = BETAO
      RP = SQRT((XINLTE-XINLT)**2+(YINLTE-YINLT)**2+(ZINLTE-ZINLT)**2)
      DXS = 0.05/FLOAT(MAX0(1,IFIX(1.75/(ZINLTE-ZINLT))))
      NFLD = 20
      NFLDO = NFLD
      XMAX = 2.*REFL
C
C  DEFINE STARTING LOCATION OF ARRAYS IN BLANK COMMON
C
      IGB(2) = IGB(1)+NBODY
      IXS = IGB(2)+NBODY
      IYS = IXS+NFLD
      IZS = IYS+NFLD
      IUS = IZS+NFLD
      IVS = IUS+NFLD
      IWS = IVS+NFLD
      ISVN = IWS+NFLD
      ISKP = ISVN+NFLD
      NASHK = ISKP+NFLD-1
C
C  CALCULATE ANGLES AND Y-LOCATION FOR SHOCK TRAVERSES
C    IS IS INDEX OF LAST PHI=0 TRAVERSE AT Y=YINLT
C
      IS = (NIS+1)/2
      DANGL = PHISHK/FLOAT(NIS-IS)
      DO 40 I=1,NIS
      IF (I.LE.IS) PHINL(I)=0.0
      IF (I.GT.IS) PHINL(I)=DANGL*FLOAT(I-IS)
  40  YINL(I) = YINLT
      IF (IS.GT.1) YINL(1)=0.0
      IF (IS.GT.2) YINL(IS-1)=YCPI
      IF (IS.LE.3) GO TO 60
      DY = YCPI/FLOAT(IS-2)
      ISM2 = IS-2
      DO 50 I=2,ISM2
  50  YINL(I) = DY*FLOAT(I)
  60  CONTINUE
C
C  PRINT SUMMARY OF INLET GEOMETRY
C
      IF (IPGTO) WRITE(6,709) XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE
     *   ,RVIVO,XCLOSD
C
C  GENERATE SET OF ZERO VALUE STRENGTHS EXCEPT ON ZERO INLET PANELS
C            INCLUDE ADDITIONAL SIDE PANELS FOR FIELD VELOCITY CALC
```

Figure A-1(r)

```
C
      NTINL = NINLET+NINBLK+NINVEL
      JGB = IGB(1)-1
      JGI = IGB(2)-1
      DO 80 I=1,NBODY
      JGB = JGB+1
      JGI = JGI+1
      A(JGI) = 0.0
      IF (INLTST(I,OPEN)) A(JGI)=A(JGB)
80    CONTINUE
C
C  ****************************************************************
C  START LOOP ON NUMBER OF MERIDIAN LOCATIONS OF TRAVERSES
C
      DO 200 IRAD=1,NIS
      NFLD = NFLD0
      PHISI = PHINL(IRAD)
      DR = 0.5*RP
C
C  MODIFY NOSE SHOCK ANGLE SHAPE TO INCLUDE ANGLE OF ATTACK VARIATION
C  REF% KRASNOV, AERODYNAMICS OF BODIES OF REVOLUTION, EQ. 7.40.
C  PSI  IS THE ANGLE OF THE POLAR ANGLE OF THE SHOCK SEARCH RELATIVE
C    TO THE INCIDENT FLOW.  AT ALPHA=0, IT IS MEASURED FROM THE
C    BOTTOM CENTERLINE.
C  BTNOSE IS TANGENT OF 90-NOSE SHOCK ANGLE PLUS ALPHA CORRECTION
C  DTHETN IS ANGLE OF ATTACK CORRECTION TO NOSE SHOCK ANGLE.
C
      THETNS = WDGSHK*DTOR
      ALP = ALPHAC*DTOR
      PSI = (PHISI-PHIR)*DTOR
      DTHETN = ALP*(1.-EALPI)*COS(PSI)
C                  90 DEG.
      BTNOSE = TAN(1.5707963-THETNS+DTHETN)
      TANALP = TAN(ALP*COS(PSI))
      BETINF = TANAPB(BETA0,TANALP)
      FML = SQRT((TANAPB(BTNOSE,-TANALP))**2+1.)
C             INITIALIZE INCREMENTS IN R AND X FOR INTEGRATION
      SINT = SIN(PHISI*DTOR)
      COST = COS(PHISI*DTOR)
      BSAVE = TANAPB(.1,TANALP)
      ANGNU1 = 2.4495*ATAN(0.40825*BETA0)
      ANGNU1 = (ANGNU1-ATAN(BETA0))*RTOD
C
C  RADIAL LOCATION LOOP ---------------------------------------------
C  FIRST X AND R LOCATED ALONG LEADING EDGE OF RAMP INLET
C
      UWR = 0.0
      VWSAV = 0.0
      ISHK2 = 2
      NINL(IRAD) = 1
      ISHK0 = MAXSHI*(IRAD-1)
      ISHK = ISHK0+1
      XINL(ISHK)=0.0
      RINL(ISHK)=0.0
      IF (IPGTO) WRITE(6,701) PHISI,XINLT,YINL(IRAD),ZINLT
      IF (IPGTO) WRITE(6,702) NINL(IRAD),XINL(ISHK),RINL(ISHK),UWR,VWSAV
     *   ,FML,BTNOSE
C
C  UNDER INLET, SHOCK PROPAGATES ALONG BTINLT TO MASS FLOW RATIO
C  SEPARATION LINE, TURNS TO NORMAL SHOCK TO POINT BELOW INLET
C  TRAILING EDGE LIP
C
      IF (PHISI.GT.0.05) GO TO 90
      ISHK = ISHK+1
      ISHK2 = 3
      NINL(IRAD) = 2
      ANGBT = ATAN(BTINLT)
      XINL(ISHK) = RP*RVIVO*SIN(ANGBT)
      R = RP*RVIVO*COS(ANGBT)
      Y = YINL(IRAD)+R*SINT
      Z = ZINLT-R*COST
      RINL(ISHK) = R
      IF (IPGTO) WRITE(6,702) NINL(IRAD),XINL(ISHK),RINL(ISHK),UWR,VWSAV
     *   ,FML,BTNOSE
      IF (RVIVO.GE.1.) GO TO 100
C
C  NORMAL SHOCK, LIMITED BY ANGLE OF ATTACK
C
      ISHK = ISHK+1
      ISHK2 = 4
      NINL(IRAD) = 3
      XINL(ISHK) = XINL(ISHK-1)+RP*(1.-RVIVO)*SIN(ALP)
      R = R+RP*(1.-RVIVO)*COS(ALP)
      Y = YINL(IRAD)+R*SINT
      Z = ZINLT-R*COST
      RINL(ISHK) = R
      IF (IPGTO) WRITE(6,702) NINL(IRAD),XINL(ISHK),RINL(ISHK),UWR,VWSAV
     *   ,FML,BTNOSE
      GO TO 100
90    CONTINUE
C
C  FOR TRAVERSES OUTSIDE INLET SIDE EDGES, COMPUTE BETA AT ALL STATIONS
C
      R = 0.0
100   R = R+DR
      ISHK = ISHK+1
      Y = R*SINT+YINL(IRAD)
      Z = -R*COST+ZINLT
      X = 1.00001*R*BTINLT+XINLT-DXS
      IF (Y.LT.YCPI) X=1.00001*SQRT(R*R+(YCPI-Y)**2)*BTINLT+XINLT-DXS
C
C  DEFINE LOCATION OF TRAVERSE TO FIND MAXIMUM CROSS FLOW VELOCITY------
C  AFTER NOSE SHOCK.
      DO 110 I=1,NFLD
      IM1 = I-1
      JXS = IXS+IM1
      JYS = IYS+IM1
      JZS = IZS+IM1
      A(JXS) = X
      A(JYS) = Y
      A(JZS) = Z
110   X = X+DXS
C
C  COMPUTE VELOCITIES AT FIELD POINTS DUE SOLELY TO INLET PANELS
C
      CALL FLDVEL (A(IXS),A(IYS),A(IZS),A(ISVN),A(ISKP)
     *   ,A(IUS),A(IVS),A(IWS),NFLD,NX,2)
C
C  SCAN NET CROSS FLOW VELOCITIES TO FIND PEAK VALUE
C
      IUSV = 0
      ISAV = 0
      UWR = 1.0
      VWR = 0.0
      WWR = 0.0
      VWSAV = 0.0
      DO 150 I=1,NFLD
      IM1 = I-1
      JUS = IUS+IM1
      JVS = IVS+IM1
      JWS = IWS+IM1
      USI = A(JUS)
      VSI = A(JVS)
      WSI = A(JWS)
      VWMAG = 0.
C             SKIP TRIVIAL CASE OF V=0,W=0
      IF (VSI.EQ.0. .AND. WSI.EQ.0.) GO TO 140
C             LIMIT SEARCH TO REGION OF COMPRESSION (U.LT.0)
      IF (USI.GT.0. .AND. IUSV.GT.0) GO TO 145
      IUSV = 1
      VWMAG = SQRT(VSI*VSI+WSI*WSI)
C             TEST FOR MAXIMUM OF RADIAL COMPONENT
      IF (VWMAG.LE.VWSAV) GO TO 140
      VWSAV = VWMAG
      ISAV = I
C
C  RESOLVE PERTURBATION VELOCITIES INTO WIND COORDINATES
C
```

Figure A-1(s)

```
      VWR = VSI*CPHI+WSI*SPHI
      WWR = -USI*SINAC-VSI*COSAC*SPHI+WSI*COSAC*CPHI
      UWR =  USI*COSAC-VSI*SINAC*SPHI+WSI*SINAC*CPHI+1.0
140   CONTINUE
C
145   JXS = IXS+IM1
      XSI = A(JXS)
150   CONTINUE
C         IF MORE POINTS COMPUTED THAN NECESSARY, RESET TO LOWER VALUE+5
      IF (ISAV.LT.NFLD) NFLD=MIN0(ISAV+5,NFLD,IUSV)
C
C  COMPUTE INCREMENTAL CHANGE IN PRANDTL-MEYER TURNING ANGLE, DNU-------
C
      VW2 = VWR*VWR+WWR*WWR
      TANNU = SQRT(VW2)/UWR
      DNU= ATAN(TANNU)/DTOR
      VW2 = VW2+UWR*UWR
C              TEST FOR EXPANSION IN FLOW
      IF (VW2.GT.1.0) DNU=-DNU
      ANGNU2 = ANGNU1-DNU
      SMALL = ANGNU2.LT.0.1
      IF (SMALL) GO TO 160
C
C  CALCULATE NEW MACH NUMBER AND BETA
C
      GO TO 170
C              IF SMALL, SET TO BSAVE VALUE
160   BETANU = BSAVE
      FNUMCH = FML
      GO TO 180
C              COMPUTE NEW MACH NUMBER
C              REFERENCE.  HALL, ROYAL AERONAUTICAL SOCIETY, SEPT 1975
170   YNU = (ANGNU2/130.450769)**(2./3.)
      YNUSQ = YNU*YNU
      YNUC  = YNU*YNUSQ
      FNUMCH =((1.+1.3604*YNU+0.0962*YNUSQ-0.5127*YNUC)
     *   /(1.-0.6722*YNU-0.3278*YNUSQ))*COS(DNU*DTOR)
      SMALL = FNUMCH.LT.1.01
      IF (SMALL) GO TO 160
      BETANU = SQRT(FNUMCH*FNUMCH-1.)
      BETANU = TANAPB(BETANU,TANALP)
      IF (BETANU.GT.BTNOSE) GO TO 180
C              SET BETANU TO LIMIT AT NOSE IF GREATER THAN NOSE VALUE
      BETANU = BTNOSE
      FNUMCH = FML
C
C  COMPUTE SHOCK SHAPE BY INTEGRATION OF NEW MACH ANGLE
C
180   BAVG = 0.5*(BETANU+BSAVE)
      NINL(IRAD) = ISHK-ISHK0
      IF (NINL(IRAD).EQ.ISHK2) BAVG=BETANU
      XINL(ISHK) = XINL(ISHK-1)+BAVG*(R-RINL(ISHK-1))
      RINL(ISHK) = R
      DXDR = (XINL(ISHK)-XINL(ISHK-1))/(R-RINL(ISHK-1))
      IF (IPGT0) WRITE(6,702) NINL(IRAD),XINL(ISHK),R,UWR,VWSAV,FNUMCH
     *   ,BETANU,DXDR,DNU,VW2
C
C  CONVERGENCE TESTS
C
      IF (XINL(ISHK).GE.XMAX) GO TO 200
      IF (ABS(BETINF-BETANU)/BETINF .LE.0.01)  GO TO 200
      IF (.NOT.SMALL .AND. ABS(BETANU-BSAVE).LT.0.025) DR=DR+DR
      BSAVE = BETANU
      FML = FNUMCH
      IF (ISHK.GE.MAXSHI*IRAD) GO TO 200
      GO TO 100
200   CONTINUE
      RETURN
      END
```

```
      LOGICAL FUNCTION INLTST(I,OPEN)
C
C  FUNCTION TO COMPARE PANEL NUMBER, I,  WITH TABLE OF INLET PANELS.
C  VALUE RETURNED FOR FUNCTION IS ,
C        .TRUE.,  IF I IS INLET PANEL
C        .FALSE., IF I IS NOT INLET PANEL OR INLET NOT EXIST
C  VALUE OF OPEN DETERMINES WHETHER PANEL IS FULLY OPEN (TRUE)
C  OR FULLY BLOCKED TO FLOW (FALSE).
C
      LOGICAL INLET,OPEN
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI
     *   ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BOPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BOOL,IZ1(12)
      IF (INLET) GO TO 100
C              NO INLET EXISTS
      INLTST = .FALSE.
      OPEN = .FALSE.
      RETURN
C              COMPARE PANEL NUMBERS
100   DO 110 J=1,NTINL
      IF (I.EQ.JINLT(J)) GO TO 120
110   CONTINUE
      INLTST = .FALSE.
      OPEN = .FALSE.
      RETURN
120   INLTST = .TRUE.
      OPEN = J.LE.NINLET
      RETURN
      END
```

```
      SUBROUTINE INLXYZ (XIN,YIN,ZIN,FAREA,OPEN)
C
C  ROUTINE TO SCAN X,Y,Z OF PANEL CORNERS TO FIND CORNER
C  TO BE USED FOR INLET SHOCK.  LOOK FOR LEADING AND TRAILING EDGES
C
      LOGICAL OPEN
      DIMENSION XIN(4),YIN(4),ZIN(4)
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI
     *   ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BINSHK/ NIS,XISHLD,EALPI,XCLOSD,MAXSHI,NINL(8),PHINL(8)
     *   ,YINL(8),XINL(80),RINL(80)
C
      DATA ISTART/0/
      IF (ISTART.GT.0) GO TO 100
C
C  INITIALIZE VALUES TO FIRST CORNER
C
      XINLT = XIN(4)
      YINLT = YIN(4)
      ZINLT = ZIN(4)
      XINLTE = XIN(3)
      YINLTE = YIN(3)
      ZINLTE = ZIN(3)
      YCPI = YIN(1)
      ISTART = 1
      RVIVO = 1.0
      AOPEN = 0.0
      ATOT = 0.0
      NTINL = NINLET+NINBLK
      XCLOSD = XINLTE
      IF (.NOT.OPEN) XCLOSD=XIN(1)
100   CONTINUE
C
C  COMPARE CORNERS 1 AND 4 TO FIND INLET OUTBOARD LEADING EDGE
C
      IF (XIN(1).GT.XINLT) GO TO 120
      IF (YIN(1).LT.YINLT) GO TO 120
      XINLT = XIN(1)
      YINLT = YIN(1)
```

Figure A-1(t)

```
      ZINLT = ZIN(1)
120   IF (XIN(4).GT.XINLT) GO TO 140
      IF (YIN(4).LT.YINLT) GO TO 140
      XINLT = XIN(4)
      YINLT = YIN(4)
      ZINLT = ZIN(4)
C
C  COMPARE CORNERS 2 AND 3 TO FIND INLET OUTBOARD TRAILING EDGE
C
140   IF (XIN(2).LT.XINLTE) GO TO 160
      XINLTE = XIN(2)
      ZINLTE = ZIN(2)
      IF (YIN(2).LT.YINLTE) GO TO 160
      YINLTE = YIN(2)
160   IF (XIN(3).LT.XINLTE) GO TO 180
      XINLTE = XIN(3)
      ZINLTE = ZIN(3)
      IF (YIN(3).LT.YINLTE) GO TO 180
      YINLTE = YIN(3)
180   CONTINUE
C
C  FIND INBOARD MOST CORNER OF INLET PANELS
C
      YCPI = AMIN1(YCPI,YIN(1),YIN(2),YIN(3),YIN(4))
C
C  RATIO OF FRONTAL AREA OF PANELS COMPLETELY OPEN TO FLOW TO TOTAL
C  FRONTAL AREA OF INLET PANELS
C
      IF (OPEN) AOPEN=AOPEN+FAREA
      ATOT = ATOT+FAREA
      RMASS = AOPEN/ATOT
      RVIVO = RMASS
C
C  FIND LEADING EDGE OF BLOCKED INLET PANELS
C
      IF (      OPEN) XCLOSD = XINLTE
      IF (.NOT.OPEN) XCLOSD=AMIN1(XCLOSD,XIN(1),XIN(4))
      RETURN
      END


      SUBROUTINE INVERT(A,NSYS,N,NMAX,AMAX)
C     SUBROUTINE TO SOLVE SIMULTANEOUS EQUATIONS
C
      DIMENSION A(NMAX,NMAX),X(200)
  5   SIGN=1.0
      NPI=N+1
      NMI=N-1
      NPLSY=N+NSYS
  9   DO 14 I=1,NMI
      IPI=I+1
      MAX=I
      AMAX=ABS(A(I,I))
      DO 10 K=IPI,N
      AKMAX=ABS(A(K,I))
      IF(AKMAX.LE.AMAX) GO TO 10
      MAX=K
      AMAX=AKMAX
 10   CONTINUE
      IF(AMAX.LT.1.0E-12) GO TO 16
      IF(MAX.EQ.I) GO TO 12
      DO 11 L=I,NPLSY
      TEMP=A(I,L)
      A(I,L)=A(MAX,L)
 11   A(MAX,L)=TEMP
      SIGN=-SIGN
 12   DO 14 J=IPI,N
      IF (A(J,I)) 30,14,30
  30 CONST=-A(J,I)/A(I,I)
      DO 13 L=1,NPLSY
 13   A(J,L)=A(J,L)+A(I,L)*CONST
 14   CONTINUE
      DO 15 I=1,N
      IF (A(I,I)) 15,16,15
  15 CONTINUE
      GO TO 18
  16 WRITE(6,100)
 100 FORMAT(///5X,24H***** MATRIX IS SINGULAR)
      STOP
  18 DO 21 I=NPI,NPLSY
      DO 20 KK=1,N
      K=NPI-KK
      X(K)=A(K,I)
      IF(K.EQ.N) GO TO 20
      J=K
  19 J=J+1
      X(K)=X(K)-A(K,J)*X(J)
      IF(J.NE.N) GO TO 19
  20 X(K)=X(K)/A(K,K)
      DO 21 J=1,N
  21 A(J,I)=X(J)
      RETURN
      END


      SUBROUTINE IOREAD (IO,A,NA)
C
C  PERFORM AN UNFORMATTED READ FROM FILE  IO  OF ARRAY  A.
C     ******* CDC VERSION FOR BUFFERED I/O ******************************
C           USE OF BUFFERED IO STATEMENTS HERE ALSO ALLOWS THE
C           USER TO SET THE FILE BUFFER LENGTH TO ZERO IN THE
C           PROGRAM CARD (I.E.   TAPE8=0)
      DIMENSION A(NA)
C
C66   FORMAT (//,27H  IOREAD PARITY ERROR - IO=,I5,5H  NA=,I5
C    *    ,9H  UNITIO=,I5)
C67   FORMAT (//,26H  IOREAD END-OF-FILE - IO=,I5,5H  NA=,I5
C    *    ,9H  UNITIO=,I5)
C     IF (      IO.EQ.12) READ (IO) A
C     IF (.NOT.IO.EQ.12) BUFFER IN (IO,0) (A(1),A(NA))
C     UNITIO = UNIT(IO)
C     IF (UNITIO) 999,967,966
C966  WRITE (6,66) IO,NA,UNITIO
C     STOP 66
C967  WRITE (6,67) IO,NA,UNITIO
C     STOP 67
C999  RETURN
C
C     ******* ANSI FORTRAN VERSION ***************************************
C              TO USE, REMOVE  C  IN COLUMN ONE AND REPLACE ABOVE CARDS
      READ (IO) A
      RETURN
C
      END


      SUBROUTINE IOWRIT(IO,A,NA)
C
C  PERFORM AN UNFORMATTED WRITE TO FILE  IO  OF ARRAY A.
C     ******* CDC VERSION FOR BUFFERED I/O ******************************
C           USE OF BUFFERED IO STATEMENTS HERE ALSO ALLOWS THE
C           USER TO SET THE FILE BUFFER LENGTH TO ZERO IN THE
C           PROGRAM CARD (I.E.   TAPE8=0)
      DIMENSION A(NA)
C
C66   FORMAT(14H  IOWRIT ERROR,2I5,G12.4)
C
C     IF (      IO.EQ.12) WRITE (IO) A
C     IF (.NOT.IO.EQ.12) BUFFER OUT (IO,0) (A(1),A(NA))
C     UNITIO = UNIT(IO)
C     IF (UNITIO.GE.0.) GO TO 999
C     RETURN
```

Figure A-1(u)

```
C***  WRITE (6,66) IO,NA,UNITIO
C     STOP 66
C
C     ******* ANSI FORTRAN VERSION ***********************************
C             TO USE, REMOVE  C  IN COLUMN ONE AND REPLACE ABOVE CARDS
C
      WRITE(IO) A
      RETURN
      END


      SUBROUTINE NETCLC(S,DP,NCW,NCW1,MSW,MSW1,MSWM1,PSILE,PSITE,
     1          PHI)
C
C     THIS ROUTINE CALCULATES THE NET STRENGTHS OR THICKNESS
C     SLOPES ASSOCIATED WITH CORNER POINTS AFTER SUPERPOSITION.
C
      DIMENSION S(NCW,MSW),DP(NCW1,MSWM1),PSILE(MSW1),PSITE(MSW1),
     1     PHI(MSW1)
      COMMON/WPGEOM/P(2400)
C
C     SET P ARRAY TO ZERO
C
      NCW2=NCW+2
      MAXP=(NCW2)*(MSWM1+1)
      DO 5 I=1,MAXP
    5 P(I)=0.0
C
C     STORE S ARRAY IN P, LEAVING A ZERO COLUMN WHENEVER A
C     BREAK IN SWEEP OR DIMEDRAL OCCURS
C
      JCOL=1
      DO 15 J=1,MSW
      JJ=JCOL*NCW2+1
      JCOL=JCOL+1
      DO 10 I=1,NCW
      IJ=JJ+I
   10 P(IJ)=S(I,J)
      IF (J.EQ.MSW) GO TO 18
      IF(PSILE(J+2).NE.PSILE(J+1).OR.PSITE(J+2).NE.PSITE(J+1).OR.
     1     PHI(J+2).NE.PHI(J+1)) JCOL=JCOL+1
   15 CONTINUE
C
C     CALCULATE NET STRENGTH/THICKNESS
C
   18 CONTINUE
      DO 20 J=1,MSWM1
      DO 20 I=1,NCW1
      K1=(J-1)*NCW2+I
      K2=J*NCW2+1
   20 DP(I,J)=P(K1)-P(K1+1)-P(K2)+P(K2+1)
C
      RETURN
      END


      SUBROUTINE NEWRAD(IFU)
C
C  REVISE BODY (FUSELAGE) MERIDIAN LINE SPACING
C  BODY DEFINATION OPTIONS FOR EACH SEGMENTS
C
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSBY(51)
     *    ,FUSAZ(51),XJ(51),PHIK(33)
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPHT(5),IUVW,XSTART,XMLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,ROOD,IZ1(12)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKW,MAXNX,MAXKR,NATOT
     *    ,NBODY,NFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NRDIM,NG,IXPT,IYPT,IZPT,ITM,ICEL,NTAP7
     *    ,IAM,IAN,IUM,ISB(7),IVB,IC,IV,IW,IVA,IWA,ICP,IPHI,IYR
     *    ,NAG,NAP,NAV,NAS,NASMX,NAFLD,IAD,ICD,IS*0,IZZ(4),NWING,IROW(50)
      COMMON          A(12000)
      IYB = 1+MAX0(2*NXNR,NG)
      IF (IFU.EQ.1) INX=1
      IF (IFU.EQ.1) ISF=1
C
C   J2TEST = 3 AND KRAD = 0 INDICATE AN ARBITRARY FUSELAGE WITH
C   MERIDIAN LINES DEFINED BY NRAD IN THE GEOMETRY INPUT
C
      KRAD = IABS(KRADX(IFU))
C  IF KTEST=0, CIRCULAR OR COMPUTE NEW Y,Z FOR FUSELAGE
C  IF KTEST=1, ARBITRARY BODY, USE ORIGINAL Y,Z*S
      KTEST=0
      IF (J2TEST.EQ.3.AND.KRAD.EQ.0) KTEST=1
      IF (KRAD.EQ.0) KRAD=NRADX(IFU)
      KRADX(IFU) = KRAD
      NYZ = NFORX(IFU)*KRAD
      IF (IXZSYM.GT.0) NYZ=NYZ+NFORX(IFU)*(KRAD-1)
      IZB = IYB+NYZ
      CALL NEWRD2(A(IYB),A(IZB),PHIK,ZFUS(INX),FUSBY(INX)
     *    ,FUSAZ(INX),A(ISF),NFORX(IFU),KRADX(IFU),NRADX(IFU)
     *    ,IXZSYM,KTEST)
C
C  SAVE YB AND ZB FOR EACH SEGMENT
C
      IF (NFUS.GT.1) CALL IOWRIT (8,A(IYB),2*NYZ)
      ISF = ISF+2*KRADX(IFU)*NFORX(IFU)
      INX = INX+NFORX(IFU)
      RETURN
      END


      SUBROUTINE NEWRD2(YB,ZB,PHIK,ZFUS,FUSBY,FUSAZ,SFUS
     *    ,NFUSOR,KRAD,NRAD,IXZSYM,KTEST)
C
C  ROUTINE TO PERFORM RADIAL INTERPOLATION IN INPUT GEOMETRY
C  ON A GIVEN BODY SEGMENT.
C     J2TEST = 1, CIRCULAR/ELLIPTIC FUSELAGE WITH NO BODY CAMBER
C            = 2, CIRCULAR/ELLIPTIC FUSELAGE WITH BODY CAMBER
C            = 3, ARBITRARY Y,Z CROSS SECTION
C     IXZSYM = 0, SYMMETRIC GEOMETRY AND LOADS ASSUMED
C            = 1, GENERATE FULL SYMMETRIC CONFIGURATION FROM HALF BODY
C            = -1, FULL CONFIGURATION AROUND BODY IS INPUT
C
      DIMENSION YB(NFUSOR,KRAD),ZB(NFUSOR,KRAD),PHIK(KRAD)
     *    ,ZFUS(NFUSOR),FUSBY(NFUSOR),FUSAZ(NFUSOR)
     *    ,SFUS(NRAD,2,NFUSOR)
     *    ,ANSIN(33),ANCOS(33),PHIN(33),YF(33),ZF(33)
C
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPHT(5),IUVW,XSTART,XMLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,ROOD,IZ1(12)
C
      XIN(X1,Y1,X2,Y2,Y)=X1+(X2-X1)*(Y-Y1)/(Y2-Y1)
C
      DATA RADD,EPS/0.01745329,1.E-6/
      EP2=EPS*EPS
      DO 30 K=1,KRAD
      IF (J2TEST.EQ.3) GO TO 30
      PHIK(K) = PHIK(K)*RADD
C                          1.5 * PI
      PHIR4 = PHIK(K)+4.712388981
      ANSIN(K)=SIN(PHIR4)
      ANCOS(K)=COS(PHIR4)
   30 CONTINUE
C
C  COMPUTE INTERMEDIATE VALUES AROUND CIRCUMFERENCE ---------------------
C               NEARIAL STATION INDEX
      DO 100 N=1,NFUSOR
      IF (J2TEST.EQ.3) GO TO 50
```

Figure A-1(v)

```
C
C   CASE 1. COMPUTE SECTION Y + Z COORDINATES FOR CIRCULAR/ELLIPTIC BODY
C
      BY=FUSBY(N)
      AZ=FUSAZ(N)
      CAM=ZFUS(N)
      DO 40 K=1,KRAD
      RAD=0.0
      IF (BY.NE.0.)
     *   RAD=1.0/SQRT((ANCOS(K)/BY)**2+(ANSIN(K)/AZ)**2)
      YB(N,K)=RAD*ANCOS(K)
      ZB(N,K)=RAD*ANSIN(K)+CAM
40    CONTINUE
      GO TO 100
C
C   CASE2. COMPUTE SECTION Y AND Z ORDINATES FOR NONCIRCULAR BODY
C    BY LINEAR INTERPOLATION
C                K,NN=CIRCUMFERENTIAL STATION INDICES
C
50    KI=2
      PHIN(1)=0.
      YB(N,1)=SFUS(1,1,N)
      ZB(N,1)=SFUS(1,2,N)
      YF(1)=YB(N,1)
      ZF(1)=ZB(N,1)
      ZC=(SFUS(1,2,N)+SFUS(NRAD,2,N))/2.
      DO 90 NN=2,NRAD
      IF (KTEST.EQ.1) GO TO 80
      YF(NN)=SFUS(NN,1,N)
      ZF(NN)=SFUS(NN,2,N)-ZC
      NM1=NN-1
      IF (YF(NN).EQ.0..AND.ZF(NN).EQ.0.) GO TO 80
      IF (ABS(YF(NN)).LE.EP2) YF(NN)=0.
      PHIN(NN)=ATAN2(YF(NN),-ZF(NN))
      DO 60 K=KI,KRAD
      IF (PHIK(K).GT.PHIN(NN)) GO TO 70
      YB(N,K)=XIN(YF(NM1),PHIN(NM1),YF(NN),PHIN(NN),PHIK(K))
      ZB(N,K)=XIN(ZF(NM1),PHIN(NM1),ZF(NN),PHIN(NN),PHIK(K))+ZC
60    CONTINUE
70    KI=K
      GO TO 90
C
C   CASE 3. USE INPUT Y,Z VALUES
C
80    YB(N,NN)=SFUS(NN,1,N)
      ZB(N,NN)=SFUS(NN,2,N)
90    CONTINUE
100   CONTINUE
C
C   GENERATE SYMMETRIC HALF OF BODY FOR IXZSYM=1 ------------------------
C   IF IXZSYM=0, NOT SYMMETRIC.  IF IXZSYM=-1, INPUT BOTH SIDES OF CONFIG
C
      IF (IXZSYM.LE.0) GO TO 106
      DO 106 N=1,NFUSOR
      K=KRAD
      DO 104 I=2,KRAD
      K=K+1
      IK=KRAD-I+1
      YB(N,K)=-YB(N,IK)
      ZB(N,K)=ZB(N,IK)
104   CONTINUE
C
      KRAD=K
106   CONTINUE
      RETURN
      END



      SUBROUTINE NULYT
C
C     THIS ROUTINE SELECTS SUBSETS OF PANEL LAYOUT DATA
C      FOR INPUT TO PROGRAM 2
C
      COMMON/[illegible]/ ... SNT2(20),CST2(20),ZLC(200),ZRC(200),THT1(200),SNGW
     1   [illegible] ...,NCTWS,LVSWP,NWBCR
      COMMON/CONST/[illegible] ...,FOURPI
      COMMON/[illegible]/ ... IP,LTP(200)
      COMMON/INDEX/NCW,MSW,... NPANLS,NCWB,NBDCH1,NBDCH2,NHD,NRIP,MP,
     1   NCP,MSP,NIP,NS,NPW,NPT
      COMMON/NUMBER/ ... NRW,NRP,NPTW,NPTP,NPTB,NPTOT,NPTT,
     1      NRWS,NRPS,NPTWS,[illegible],NPTOTS,NPTTS,NRWP,NPTWP,
     2      NCW1,NCP1,NCWB1,NCWS1,NCPS1
      COMMON/PYGEOM/Z(20),[illegible],CHP,HP,PSIPLE(20),PSIPTE(20),IP,
     1   SLLE,PSLEDF,CENTER,ZFL,LVSPP
      COMMON/THREAT/NCWS,NCPS,MSWS,MPS,MSWS,MSPS,NTHP,ARFT(400),ARRT(400),
     1   XLFT(400),[illegible],ZLCT(400),ZHCT(400),ZLCT(400),
     2   THETAL(400),THETAT(400),SCLET(400),SLTET(400),DZCB(400),
     3   YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4   PSPSTE(20),SNPHS(400),CSPHS(400)
      COMMON /WGEOM / [illegible],CHW,SLPWLE,SLPWTE,PSIWLE(20),
     1      PSIWTE(20),Y(20),PHI(20),ZDIHED,WICR
      COMMON/WPGEOM/[illegible](200),[illegible](200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),[illegible](200),[illegible](200),ZLPT(200),SWPPLE(200),SWPPTE(200),
     2   SNPHI(200),CSPHI(200)
      COMMON            XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
     1      SWP(50),SNWP(40),CSWP(40),THTBP(40),DPNET(500),
     2      XPTS(100),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
     3      SWPS(1000),THTNET(1000)
C
      DIMENSION PHI1(20)
      DIMENSION A(7500)
      EQUIVALENCE (XPT(1),A(1))
C
C     COUNT NUMBER OF BREAKS IN SWEEP/DIHEDRAL ON WING AND PYLON.
C     COMPUTE INDICES NEEDED TO DESCRIBE NEW ARRAYS
C
      KW=0
      IF(MSWS.EQ.1)GO TO 10
      DO 5 I=2,MSWS
      IF(PSWSLE(I+1).NE.PSWSLE(I).OR.PSWSTE(I+1).NE.PSWSTE(I)
     1     .OR.PHIS(I+1).NE.PHIS(I)) KW=KW+1
    5 CONTINUE
   10 MSW1=MSW+1
      NCW1=NCW+1
      MSWS1=MSWS+1
      NCWS1=NCWS+1
      NRW=MSW1+KW
      NPTW=NCW1*NRW
      NRWS=MSWS1+KW
      NPTWS=NCWS1*NRWS
C
      NRP=0
      NPTP=0
      NRPS=0
      NPTPS=0
      IF(NPY.EQ.0)GO TO 17
      KP=0
      IF(MSPS.EQ.1)GO TO 16
      DO 15 I=2,MSPS
   15 IF(PSPSLE(I+1).NE.PSPSLE(I).OR.PSPSTE(I+1).NE.PSPSTE(I))KP=KP+1
   16 MSP1=MSP+1
      NCP1=NCP+1
      MSPS1=MSPS+1
      NCPS1=NCPS+1
      NRP=MSP1+KP
      NPTP=NCP1*NRP
      NRPS=MSPS1+KP
      NPTPS=NCPS1*NRPS
C
   17 NRB=0
      NPTR=0
      IF(NFU.EQ.0) GO TO 20
      NCWB1=NCWB+1
      NRB=2*NBD
      NPTR=NCWB1*NRB
   20 NRWP=NRW+NRP
```

Figure A-1(w)

```
      NPTWP=NPTW+NPTO
      NPTOT=NPWP+NPB
      NPTT  =NPTWP+NPTB
      NPTOTS=NPWS+NPPS
      NPTTS =NPTWS+NPTPS
C
C     LAYOUT DATA FOR CONSTANT U-VELOCITY PANELS
C
C          WING
C
      JC=0
      DO 40 JP=1,NSW
      JJP=(JP-1)*NCW
      JJP1=JJP+1
      JJC=JC*NCW1
      JC=JC+1
C
C          CORNERS ON RIGHT SIDE OF PANEL
C
      YPT(JC)=YRC(JJP1)
      ZPT(JC)=ZRC(JJP1)
      SPHI(JC)=SNPHI(JJP1)
      CPHI(JC)=CSPHI(JJP1)
      DO 25 I=1,NCW
      IP=JJP+I
      IC=JJC+I
      XPT(IC)=XRF(IP)
   25 SWP(IC)=SWPPLE(IP)
      XPT(IC+1)=XRB(IP)
      SWP(IC+1)=SWPPTE(IP)
C
      IF(JP.EQ.NSW)GO TO 30
      IF(PSIWLE(JP+2).EQ.PSIWLE(JP+1).AND.PSIWTE(JP+2).EQ.PSIWTE(JP+1)
     1   .AND.PHID(JP+2).EQ.PHID(JP+1))GO TO 40
C
C          CORNERS ON LEFT SIDE OF PANEL
C
   30 JJC=JC*NCW1
      JC=JC+1
      YPT(JC)=YLC(JJP1)
      ZPT(JC)=ZLC(JJP1)
      SPHI(JC)=SNPHI(JJP1)
      CPHI(JC)=CSPHI(JJP1)
      DO 35 I=1,NCW
      IC=JJC+I
      IP=JJP+I
      XPT(IC)=XLF(IP)
   35 SWP(IC)=SWPPLE(IP)
      XPT(IC+1)=XLB(IP)
      SWP(IC+1)=SWPPTE(IP)
   40 CONTINUE
C
C          PYLON
C
      IF(NPY.EQ.0)GO TO 65
      JC=0
      DO 60 JP=1,NSP
      JJP=(JP-1)*NCP+NPANLS
      JJP1=JJP+1
      JJC=JC*NCP1+NPTW
      JC=JC+1
       JCNRW=JC+NRW
      YPT(JCNRW)=YRC(JJP1)
      ZPT(JCNRW)=ZRC(JJP1)
      DO 45 I=1,NCP
      IP=JJP+I
      IC=JJC+I
      XPT(IC)=XRF(IP)
   45 SWP(IC)=SWPPLE(IP)
      XPT(IC+1)=XRB(IP)
      SWP(IC+1)=SWPPTE(IP)
      IF(JP.EQ.NSP)GO TO 50
      IF(PSIPLE(JP+2).EQ.PSIPLE(JP+1).AND.PSIPTE(JP+2).EQ.PSIPTE(JP+1))
     1      GO TO 60
C
   50 JJC=JC*NCP1+NPTW
      JC=JC+1
      JCNRW=JC+NRW
      YPT(JCNRW)=YLC(JJP1)
      ZPT(JCNRW)=ZLC(JJP1)
      DO 55 I=1,NCP
      IP=JJP+I
      IC=JJC+I
      XPT(IC)=XLF(IP)
   55 SWP(IC)=SWPPLE(IP)
      XPT(IC+1)=XLB(IP)
      SWP(IC+1)=SWPPTE(IP)
   60 CONTINUE
C
C          BODY - PANELS BELOW WING
C
   65 IF(NFU.EQ.0)GO TO 90
      JC=-1
      KOD=1
      IF(NBDCR2.EQ.0)GO TO 85
      J1 = 1
      J2 = NBDCR2
   70 CONTINUE
      DO 80 JP=J1,J2
      JPB=JP+N2
      JC=JC+2
      JCB=JC+NRWP
      JJC=(JC-1)*NCWB1+NPTWP
      YPT(JCB)=YLC(JPB)
      ZPT(JCB)=ZLC(JPB)
      SNBP(JC)=SNT2(JP)
      CSBP(JC)=CST2(JP)
      THTBP(JC)=THTI(JP)
      YPT(JCB+1)=YRC(JPB)
      ZPT(JCB+1)=ZRC(JPB)
      SNBP(JC+1)=SNT2(JP)
      CSBP(JC+1)=CST2(JP)
      THTBP(JC+1)=THTI(JP)
      DO 75 I=1,NCWB
      IP=(I-1)*NBD+JPB
      IC=JJC+I
      XPT(IC)=XLF(IP)
      ICNCW=IC+NCWB1
   75 XPT(ICNCW)=XRF(IP)
      XPT(IC+1)=XLB(IP)
      XPT(ICNCW+1)=XRB(IP)
   80 CONTINUE
      IF(KOD.EQ.2.OR.NBDCR1.EQ.0)GO TO 86
C
C          - PANELS ABOVE WING
C
   85 J1 = NBDCR2+1
      J2 = NBD
      KOD=2
      GO TO 70
C
C         RE-ORDER BODY PANEL STRENGTHS FOR INPUT
C         TO NET-DELTP CALCULATION. USE SNT2   STORAGE
C
   86 CONTINUE
      IF(NBDCR2.EQ.0) GO TO 89
      KOD=1
      JR1=1
      JR2=NBDCR2
   87 CONTINUE
      DO 88 JROW=JR1,JR2
      JNEW=(JROW-1)*NCWB
      JOLD=JROW+N2
      DO 88 I=1,NCWB
      INEW=JNEW+I
      IOLD=(I-1)*NBD+JOLD
      SNT2(INEW)=DELTP(IOLD)
   88 CONTINUE
```

Figure A-1(x)

```
      IF(KOD.EQ.2.OR.NBDCR1.EQ.0)GO TO 90
   89 KOD=2
      JR1=NBDCR2+1
      JR2=NBD
      GO TO 87
C
C     SOURCE PANEL ARRAYS
C
C          WING
C
   90 JC=0
      DO 110 JP=1,MSWS
      JJP=(JP-1)*NCWS
      JJP1=JJP+1
      JJC=JC*NCWS1
      JC=JC+1
      YPTS(JC)=YRCT(JJP1)
      ZPTS(JC)=ZRCT(JJP1)
      SPHS(JC)=SNPHS(JJP1)
      CPHS(JC)=CSPHS(JJP1)
      DO 95 I=1,NCWS
      IP=JJP+I
      IC=JJC+I
      XPTS(IC)=XRFT(IP)
   95 SWPS(IC)=SLLET(IP)
      XPTS(IC+1)=XRBT(IP)
      SWPS(IC+1)=SLTET(IP)
C
      IF(JP.EQ.MSWS)GO TO 100
      IF(PSWSLE(JP+2).EQ.PSWSLE(JP+1).AND.PSWSTE(JP+2).EQ.PSWSTE(JP+1)
     1    .AND.PHIS(JP+2).EQ.PHIS(JP+1))GO TO 110
  100 JJC=JC*NCWS1
      JC=JC+1
      YPTS(JC)=YLCT(JJP1)
      ZPTS(JC)=ZLCT(JJP1)
      SPHS(JC)=SNPHS(JJP1)
      CPHS(JC)=CSPHS(JJP1)
      DO 105 I=1,NCWS
      IC=JJC+I
      IP=JJP+I
      XPTS(IC)=XLFT(IP)
  105 SWPS(IC)=SLLET(IP)
      XPTS(IC+1)=XLBT(IP)
      SWPS(IC+1)=SLTET(IP)
  110 CONTINUE
C
C          PYLON
C
      IF(NPY.EQ.0)GO TO 132
      JC=0
      DO 130 JP=1,MSPS
      JJP=(JP-1)*NCPS+MS
      JJP1=JJP+1
      JJC=JC*NCPS1+NPTWS
      JC=JC+1
      JCNRW=JC+NRWS
      YPTS(JCNRW)=YRCT(JJP1)
      ZPTS(JCNRW)=ZRCT(JJP1)
      DO 115 I=1,NCPS
      IP=JJP+I
      IC=JJC+I
      XPTS(IC)=XRFT(IP)
  115 SWPS(IC)=SLLET(IP)
      XPTS(IC+1)=XRBT(IP)
      SWPS(IC+1)=SLTET(IP)
      IF(JP.EQ.MSPS) GO TO 120
      IF(PSPSLE(JP+2).EQ.PSPSLE(JP+1).AND.PSPSTE(JP+2).EQ.PSPSTE(JP+1))
     1    GO TO 130
C
  120 JJC=JC*NCPS1+NPTWS
      JC=JC+1
      JCNRW=JC+NRWS
      YPTS(JCNRW)=YLCT(JJP1)
      ZPTS(JCNRW)=ZLCT(JJP1)
      DO 125 I=1,NCPS
      IP=JJP+I
      IC=JJC+I
      XPTS(IC)=XLFT(IP)
  125 SWPS(IC)=SLLET(IP)
      XPTS(IC+1)=XLBT(IP)
      SWPS(IC+1)=SLTET(IP)
  130 CONTINUE
C
C     CALCULATE NET DELTP/THICKNESS SLOPES
C
C          WING
C
  132 CONTINUE
      CALL NETCLC(DELTP,DPNET,NCW,NCW1,MSW,MSW1,NRW,PSIWLE,
     1    PSIWTE,PHID)
      CALL NETCLC(DZDX,THTNET,NCWS,NCWS1,MSWS,MSWS1,NRWS,PSWSLE,
     1    PSWSTE,PHIS)
C
C          PYLON
C
      IF(NPY.EQ.0)GO TO 140
      M=MAX0(MSP1,MSPS1)
      DO 135 I=1,M
  135 PHIP(I)=0.0
      CALL NETCLC(DELTP(NPANLS+1),DPNET(NPTW+1),NCP,NCP1,MSP,MSP1,
     1    NRP,PSIPLE,PSIPTE,PHIP)
      CALL NETCLC(DZDX(MS+1),THTNET(NPTWS+1),NCPS,NCPS1,MSPS,
     1    MSPS1,NRPS,PSPSLE,PSPSTE,PHIP)
C
C          BODY CONSTANT U PANELS
C
  140 CONTINUE
      IF(NFU.EQ.0)GO TO 145
      CALL NETCLC(SNT2,DPNET(NPTWP+1),NCWB,NCWB1,NBD,
     1    NBD+1,NRB,THTI,THTI,THTI)
C
C     RESTORE FACTOR OF PI TO THICKNESS SLOPES
C
  145 CONTINUE
      DO 150 I=1,NTHP
  150 DZDX(I)=DZDX(I)*PI
C
      RETURN
      END


      SUBROUTINE PANEL(J,K,NP,XYZ,NBODY,XC,YC,ZC,KFUSOR)
C
C   CALCULATE PANEL GEOMETRY (BASED ON THE HYPERSONIC ARBITRARY BODY
C   PROGRAM OF A. E. GENTRY)  FOR NON-PLANAR PANELS
C
      DIMENSION XC(KFUSOR),YC(KFUSOR,K),ZC(KFUSOR,K),XYZ(NBODY,6)
      DIMENSION XIN(4),YIN(4),ZIN(4),XI(4),ETA(4)
      REAL NX,NY,NZ
      LOGICAL INLTST,OPEN
      DATA EPS/1.E-6/
C
C   REORDER THE PANEL CORNER POINTS TO CORRESPOND TO GENTRY CONVENTION
C
      J1=J-1
      K1=K-1
      XIN(1)=XC(J1)
      XIN(2)=XC(J)
      XIN(3)=XC(J)
      XIN(4)=XC(J1)
      YIN(1)=YC(J1,K1)
      YIN(2)=YC(J,K1)
      YIN(3)=YC(J,K)
      YIN(4)=YC(J1,K)
      ZIN(1)=ZC(J1,K1)
      ZIN(2)=ZC(J,K1)
```

Figure A-1(y)

```
      ZIN(3)=ZC(J,K)
      ZIN(4)=ZC(J1,K)
C
C   FORM DIAGONAL VECTORS
C   FORM VECTOR CROSS PRODUCT, N = T2 X T1
C
20    T1X=XIN(3)-XIN(1)
      T2X=XIN(4)-XIN(2)
      T1Y=YIN(3)-YIN(1)
      T2Y=YIN(4)-YIN(2)
      T1Z=ZIN(3)-ZIN(1)
      T2Z=ZIN(4)-ZIN(2)
      NX=T2Y*T1Z-T1Y*T2Z
      NY=T1X*T2Z-T2X*T1Z
      NZ=T2X*T1Y-T1X*T2Y
      IF (ABS(NX).LE.EPS) NX=0.
      IF (ABS(NY).LE.EPS) N =0.
      IF (ABS(NZ).LE.EPS) NZ=0.
      VN=SQRT(NX*NX+NY*NY+NZ*NZ)
      IF (VN.EQ.0.) GO TO 30
C
C   FORM UNIT NORMAL VECTOR, THEN COMPUTE AVERAGE POINT
C
      VND=1./VN
      NX=NX*VND
      NY=NY*VND
      NZ=NZ*VND
30    AVX=.25*(XIN(1)+XIN(2)+XIN(3)+XIN(4))
      AVY=.25*(YIN(1)+YIN(2)+YIN(3)+YIN(4))
      AVZ=.25*(ZIN(1)+ZIN(2)+ZIN(3)+ZIN(4))
C
C   COMPUTE PROJECTION DISTANCE
C
      D=NX*(AVX-XIN(1))+NY*(AVY-YIN(1))+NZ*(AVZ-ZIN(1))
      T=SQRT(T1X*T1X+T1Y*T1Y+T1Z*T1Z)
      IF (T.EQ.0.) GO TO 40
      TD=1./T
      T1X=T1X*TD
      T1Y=T1Y*TD
      T1Z=T1Z*TD
40    T2X=NY*T1Z-NZ*T1Y
      T2Y=NZ*T1X-NX*T1Z
      T2Z=NX*T1Y-NY*T1X
C
C   COMPUTE COORDINATES OF CORNER POINTS IN REFERENCE COORDINATE
C   SYSTEM
C   TRANSFORM CORNER POINT TO ELEMENT COORDINATE SYSTEM (XI,ETA)
C   WITH AVERAGE POINT AS ORIGIN
C
      DO 50 N=1,4
      XPA=XIN(N)+NX*D
      YPA=YIN(N)+NY*D
      ZPA=ZIN(N)+NZ*D
      D=-D
      XDIF=XPA-AVX
      YDIF=YPA-AVY
      ZDIF=ZPA-AVZ
      XI(N)=T1X*XDIF+T1Y*YDIF+T1Z*ZDIF
     ETA(N)=T2X*XDIF+T2Y*YDIF+T2Z*ZDIF
50    CONTINUE
C
C   COMPUTE CENTROID
C   OBTAIN CORNER POINTS IN SYSTEM WITH CENTROID AS ORIGIN
C
      ETACK=ETA(2)-ETA(4)
      IF (ETACK.NE.0.) GO TO 60
      XI0=0.
      GO TO 70
60    XI0=(XI(4)*(ETA(1)-ETA(2))+XI(2)*(ETA(4)-ETA(1)))/(3.*ETACK)
70    ETA0=-ETA(1)/3.
      XI(1)=XI(1)-XI0
      XI(2)=XI(2)-XI0
      XI(3)=XI(3)-XI0
      XI(4)=XI(4)-XI0
      ETA(1)=ETA(1)-ETA0
      ETA(2)=ETA(2)-ETA0
      ETA(3)=ETA(3)-ETA0
      ETA(4)=ETA(4)-ETA0
C
C   TRANSFORM CENTROID TO REFERENCE COORDINATE SYSTEM
C
      XPT=AVX+T1X*XI0+T2X*ETA0
      YPT=AVY+T1Y*XI0+T2Y*ETA0
      ZPT=AVZ+T1Z*XI0+T2Z*ETA0
C
C   COMPUTE PANEL INCIDENCE AND INCLINATION ANGLE
C   COMPUTE PANEL AREA
C
      DELTA=0.
      THET=0.
      RN=SQRT(NY*NY+NZ*NZ)
      IF (NX.NE.0.) DELTA=ATAN2(-NX,RN)
      IF (NY.EQ.0..AND.NZ.EQ.0.) GO TO 100
      THET=ATAN2(-NY,NZ)
100   AP=.5*(XI(3)-XI(1))*ETACK
      XYZ(NP,1) = XPT
      XYZ(NP,2) = YPT
      XYZ(NP,3) = ZPT
      XYZ(NP,4) = THET
      XYZ(NP,5) = DELTA
      XYZ(NP,6) = AP
C
C  CHECK FOR INLET PANEL NUMBER AND COMPARE WITH VALUES OF LEADING
C  AND TRAILING EDGES TO FIND CORNERS FOR SHOCK
C
      IF (INLTST(NP,OPEN)) CALL INLXYZ (XIN,YIN,ZIN,AP*SIN(DELTA),OPEN)
      RETURN
      END




      SUBROUTINE PANVEL (UB,VB,WB,AN,IXZSYM)
C
C  ROUTINE TO ORGANIZE THE CALCULATION OF THE INFLUENCE OF A GIVEN
C  PANEL *J* ON A FIELD POINT (XPTI,YPTI,ZPTI).  UB,VB,WB ARE THE
C  THREE ORTHOGONAL VELOCITY INFLUENCES ASSOCIATED WITH THE BODY AXES.
C  *AN* IS THE NORMAL INFLUENCE.
C
      LOGICAL LZERO,SUBSON,SUPERS
      COMMON /BLKPAN/ COST,SINT,XBTJ,YBTJ,ZBTJ,XC1,YC1,ZC1
     *   ,XPTI,YPTI,ZPTI,COSTI,SINTI,COSD,SIND,LZERO
      COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4)
     1   ,XI,YI,ZI,XJ,ZJ,RETA0,BETAL,SUBSON,SUPERS,JP
C
C             LZERO = .FALSE. - DEFINE ANGLE TRANSFORMATIONS
C                     .TRUE.  - SKIP TRANSFORMATION CALCULATION
      IF (LZERO) GO TO 40
      XW=SINT*COSTI
      XX=COST*SINTI
      XY=COST*COSTI
      XZ=SINT*SINTI
      SINTR=XW-XX
      SINTL=XW+XX
      COSTR=XY+XZ
      COSTL=XY-XZ
C
C  CALCULATION OF CONTROL POINT IN PANEL COORDINATE SYSTEM
C
40    XI=XPTI-XC1
      DY=YPTI-YC1
      DZ=ZPTI-ZC1
      YI=DY*COST+DZ*SINT
      ZI=DZ*COST-DY*SINT
      XJ =XBTJ-XC1
      DYJ=YBTJ-YC1
      DZJ=ZBTJ-ZC1
      ZJ=DZJ*COST-DYJ*SINT
```

Figure A-1(z)

```
C
C  CALCULATE VELOCITY COMPONENTS INDUCED BY CONSTANT SOURCE
C  DISTRIBUTION PANELS
C
      CALL SORPAN(UR,VR,WR)
      UB=UR
      VI=VR*COSTR-WR*SINTR
      WI=VR*SINTR+WR*COSTR
      UL=0.
      VL=0.
      WL=0.
C
C  CALCULATE VELOCITY COMPONENTS FROM LEFT SIDE FOR SYMMETRIC
C  CONFIGURATIONS INDUCED BY CONSTANT SOURCE PANEL
C
      IF (IXZSYM.NE.0) GO TO 45
      DY=-YPTI-YC1
      YI=DY*COST+DZ*SINT
      ZI=DZ*COST-DY*SINT
      CALL SORPAN(UL,VL,WL)
      UB=UB+UL
      VI=VI-VL*COSTL+WL*SINTL
      WI=WI+VL*SINTL+WL*COSTL
45    CONTINUE
C
C  CALCULATE VELOCITY COMPONENTS IN ORIGINAL COORDINATE SYSTEM
C
      VB=VI*COSTI-WI*SINTI
      WB=WI*COSTI+VI*SINTI
      AN=WI*COSD-UB*SIND
      RETURN
      END


      SUBROUTINE PAS001 (A,N,NER,ND)
C
C  LU DECOMPOSITION OF POSITIVE DEFINITE MATRIX - A
C  ARGUMENTS%
C     A   = MATRIX OF COEFFICIENTS, A(ND,N)
C     N   = ACTUAL SIZE OF A(N,N)
C     NER = DIAGNOSTIC.  IF NER.GT.0 ON RETURN, DECOMPOSITION FAILED
C           BECAUSE ABS(A(NER,NER)) ELEMENT IS LESS THAN 1.0E-20
C     ND  = DIMENSIONED SIZE OF A.
C  UPON RETURN, A CONTAINS LU WHERE THE DIAGONAL ELEMENTS OF L ARE 1.
C  BY G.N. VANDERPLAATS                          JAN. 1973
C  NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CAL
C
      DIMENSION A(ND,N)
      NM1=N-1
      NER=0
      DO 40 K=1,NM1
      KP1=K+1
      AA=A(K,K)
      IF (ABS(AA).GT.1.0E-20) GO TO 10
C  SINGULAR MATRIX
      NER=K
      RETURN
10    AA=1.0/AA
      DO 30 I=KP1,N
      BB=-A(I,K)*AA
      IF (BB.EQ.0.0) GO TO 30
      A(I,K)=BB
      DO 20 J=KP1,N
20    A(I,J)=A(I,J)+BB*A(K,J)
30    CONTINUE
40    CONTINUE
      RETURN
      END

      SUBROUTINE PAS002 (A,B,N,NB,NDA,NDB)
C
C  SOLVES THE SYSTEM OF EQUATIONS LU*X = B, BY FORWARD AND BACKWARD
C  SUBSTITUTION.  IT IS ASSUMED THAT MATRIX, A, HAS BEEN DECOMPOSED
C  BY ROUTINE PAS001 OR EQUIVALENT SO THAT A CONTAINS LU.
C  ARGUMENTS%
C     A   = COEFFICIENT MATRIX CONTAINING LU.  A(NDA,N)
C     B   = MATRIX CONTAINING RIGHT HAND SIDES OF THE LINEAR SYSTEM
C              LU*X = B
C     N   = ACTUAL SIZE OF LU MATRIX CONTAINED IN A.
C     NB  = NUMBER OF RIGHT HAND SIDE VECTORS CONTAINED IN THE FIRST
C           NB COLUMNS OF B.
C     NDA = DIMENSIONED SIZE OF A.
C     NDB = DIMENSIONED SIZE OF B.
C  BY G.N. VANDERPLAATS                            JAN., 1973
C  NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CAL
C
      DIMENSION A(NDA,N),B(NDB,NB)
      NM1=N-1
      NP1=N+1
C
C  FORWARD SUBSTITUTION
      DO 10 K=1,NM1
      KP1=K+1
      DO 10 I=KP1,N
      AA=A(I,K)
      DO 10 J=1,NB
10    B(I,J)=B(I,J)+AA*B(K,J)
C
C  BACKWARD SUBSTITUTION
      AA=1.0/A(N,N)
      DO 20 J=1,NB
20    B(N,J)=AA*B(N,J)
      DO 50 K=2,N
      I=NP1-K
      J1=I+1
      DO 30 J=J1,N
      AA=A(I,J)
      DO 30 L=1,NB
30    B(I,L)=B(I,L)-AA*B(J,L)
      AA=1.0/A(I,I)
      DO 40 L=1,NB
40    B(I,L)=AA*B(I,L)
50    CONTINUE
      RETURN
      END


      SUBROUTINE PLYOUT
C
C     SUBROUTINE TO READ AND PRINT PYLON DATA AND LAY OUT ELEMENTAL
C     PANELS ON THE PYLON
C     PANEL COORDINATES ARE IN WING SYSTEM
C
C     CORNER ORIENTATIONS  RIGHT SIDE IS CLOSER TO THE WING
C
      COMMON/BPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THT1(200),SNGW
      COMMON/CONSTS/PI,PI2,DTOR,RTOD,FOURPI
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON/PYGEOM/Z(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
     1       SLLE,PSLPDF,CENTER,ZPL,LVSPP
      COMMON /WGEOM / XRWOC,ZRWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1       PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR
      COMMON/WPGEOM/XRF(200),XRR(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2   SNPHI(200),CSPHI(200)
C
  701 FORMAT(10I5)
  703 FORMAT(I5,7F10.0)
  704 FORMAT(1H1,10X,16HPYLON INPUT DATA)
  705 FORMAT(/15X,41HLEADING EDGE OF PYLON ROOT CHORD IS AT X=,F10.5,45H
```

Figure A-1(aa)

```
      1 FEET (MEASURED FROM LOCAL WING LEADING EDGE)/15X,23HSPANWISE LOCA
      2TION IS Y=,F10.5,5H FEET/15X,11HROOTCHORD =,F10.5,5H FEET/15X,8HHE
      3IGHT =,F10.5,5H FEET)
  706 FORMAT(/14X,I4,  50H U-VELOCITY PANELS ARE TO BE LAID OUT ON THE P
     1YLON/20X,
     2  I3,20H CHORDWISE ROWS WITH,I3,12H IN EACH ROW)
  709 FORMAT(/15X,58HSPANWISE LOCATIONS OF PANEL SIDE EDGES AND SWEEP AN
     1GLES OF/20X,24HPYLON SECTION TO THE RIGHT//
     2  20X,1HK,9X,1HZ,6X,18HLE SWEEP  TE SWEEP/
     3  28X,4HFEET,5X,17HDEGREES   DEGREES)
  710 FORMAT(18X,I3,4X,4F10.5)
C
      DATA FAC/0.95/
C
C     READ PYLON DATA
C
      READ(5,703) IP,CRP,MP,XPLE
      WRITE(6,704)
      YPL=0.0
      IF(IP.GT.0) YPL=Y(IP)
      WRITE(6,705) XPLE,YPL,CRP,MP
      READ(5,701) NCP,MSP
      MP=NCP*MSP
      WRITE(6,706) MP,MSP,NCP
      KMAX=MSP+1
      DO 1 J=1,KMAX
    1 READ(5,703) K,Z(K),PSIPLE(K),PSIPTE(K)
      WRITE (6,709)
      DO 2 K=1,KMAX
    2 WRITE (6,710) K,Z(K),PSIPLE(K),PSIPTE(K)
C
C     LAY OUT ELEMENTAL PANELS
C
      LVSPP=0
      IF(MSP.EQ.1) GO TO 6
      DO 4 I=3,KMAX
    4 IF(PSIPLE(I).NE.PSIPLE(I-1).OR.PSIPTE(I).NE.PSIPTE(I-1)) LVSPP=1
    6 SLLE=TAN(PSIPLE(KMAX)*DTOR)
      SLTE=TAN(PSIPTE(KMAX)*DTOR)
      PSLPDF=SLLE-SLTE
      IF(IP.GT.1) GO TO 5
      PLEX=XPLE
      ZPL=0.0
      GO TO 10
    5 CONTINUE
      INDX=(IP-2)*NCW+1
      PLEX=XPLE+XLF(INDX)
      ZPL=ZLC(INDX)
   10 CONTINUE
      ANCP=NCP
      CSIDE=CRP
C
C     LOOP OVER CHORDWISE ROWS OF PANELS
C
      DO 100 I=2,KMAX
      IF(LVSPP.EQ.0) GO TO 20
      SLLE=TAN(PSIPLE(I)*DTOR)
      SLTE=TAN(PSIPTE(I)*DTOR)
      PSLPDF=SLLE-SLTE
   20 IM=I-1
      DZ=Z(IM)-Z(I)
      CSIDEP=CSIDE
      CSIDE=CSIDEP+DZ*PSLPDF
      IF(I.EQ.2) GO TO 30
      JLE=(I-3)*NCP+1-NPANLS
      PLEX=XLF(JLE)
   30 CONTINUE
C
C     LOOP OVER ELEMENTAL PANELS IN THIS CHORDWISE ROW
C
      DO 130 K=1,NCP
      J=(I-2)*NCP+K+NPANLS
      AKM=K-1
      AK=K
      SWPPLE(J)=SLLE-AKM*PSLPDF/ANCP
      SWPPTE(J)=SLLE-AK*PSLPDF/ANCP
C
C     CORNER POINT COORDINATES
C
      XRF(J)=-AKM*CSIDEP/ANCP+PLEX
      XRB(J)=XRF(J)-CSIDEP/ANCP
      XLF(J)=-AKM*CSIDE/ANCP+DZ*SLLE+PLEX
      XLB(J)=XLF(J)-CSIDE/ANCP
      YLC(J)=YPL
      YRC(J)=YPL
      ZRC(J)=Z(IM)+ZPL
      ZLC(J)=Z(I)+ZPL
C
C     Z-COORDINATES OF CONTROL POINT AND PANEL CENTROID
C
      IF(K.GT.1) GO TO 40
      A1=XLB(J)-XLF(J)
      A2=XRB(J)-XRF(J)
      ZBAR=(2.0*A1+A2)*(-DZ)/(3.0*(A1+A2))
   40 CONTINUE
      ZCPT(J)=Z(IM)+ZBAR+ZPL
      YCPT(J)=YPL
C
C     X-COORDINATES OF CONTROL POINT
C
      XPCLE=XRF(J)-ZBAR*SWPPLE(J)
      XPCTE=XRB(J)-ZBAR*SWPPTE(J)
      PNLC=XPCLE-XPCTE
      XCPT(J)=XPCLE-FAC*PNLC
  130 CONTINUE
  100 CONTINUE
C
      RETURN
      END

      SUBROUTINE PYTHIN
C
C     SUBROUTINE TO INPUT PYLON THICKNESS DATA
C
      COMMON/THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     1   XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2   THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3   YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4   PSPSTE(20),SNPHS(400),CSPHS(400)
C
  701 FORMAT(10I5)
  702 FORMAT(8F10.0)
  703 FORMAT(//14X,I4,49H THICKNESS PANELS ARE TO BE LAID OUT ON THE PYL
     1ON/20X,I3,20H CHORDWISE ROWS WITH,I3,12H IN EACH ROW/
     2  22X,30HPYLON THICKNESS OPTION, NUNIP=, I2//)
  705 FORMAT(I5,4F10.0)
  706 FORMAT(15X,58HSPANWISE LOCATIONS OF PANEL SIDE EDGES AND SWEEP ANG
     1LES OF/20X,26HPYLON SECTION TO THE RIGHT//
     2  20X,1HK,9X,1HZ,6X,18HLE SWEEP  TE SWEEP/
     3  28X,4HFEET,5X,17HDEGREES   DEGREES)
  707 FORMAT(18X,I3,4X,4F10.5)
C
      READ (5,701)NCPS,MSPS,NUNIP
      MPS=NCPS*MSPS
      WRITE(6,703)MPS,MSPS,NCPS,NUNIP
      MS1=MSPS+1
      DO 1 J=1,MS1
    1 READ(5,705) I,ZS(I),PSPSLE(I),PSPSTE(I)
      WRITE(6,706)
      DO 2 J=1,MS1
    2 WRITE(6,707) J,ZS(J),PSPSLE(J),PSPSTE(J)
      IF(NUNIP .NE. 0)GO TO 5
      MN=0
      DO 3 JNP=1,MPS,NCPS
      MN=MN+NCPS
```

Figure A-1(bb)

```
    3 READ(5,702) (THETPL(JP),JP=JNP,MN)
      RETURN
    5 READ(5,702) (THETPL(JP),JP=1,NCPS)
      DO 6 J=2,MSPS
      JJ=(J-1)*NCPS
      DO 6 K=1,NCPS
      KK=JJ+K
    6 THETPL(KK)=THETPL(K)
      RETURN
      END


      SUBROUTINE RACKIO
C
C     SUBROUTINE TO INPUT AND OUTPUT RACK DATA
C          LINE SOURCE AND DOUBLET DISTRIBUTION ARE CALCULATED
C          USING SUBROUTINE BDYGEN
C
      COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7),
     1 SSUMK(7),SSUMKD(7)
      COMMON/BPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1     SANGW
      COMMON /CONSTS/PI,PI2,DTOR,RTOD,FOURPI
      COMMON /FLOW/ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ,FMCHSQ
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON/PYGEOM/Z(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
     1     SLLE,PSLPDF,CENTER,ZPL,LVSPP
      COMMON /RKGEOM/ RRMAX,RLTHC,XWROC,YWRO,ZWRO,NRPOLY,RXEND(7),
     1        RCOEF(7,7),XBRO,YBRO,ZBRO,RIBCR,SRIBCR,CRIBCR
      COMMON /RSHOCK/ NRSHK,RXSHK(50),RRSHK(50),RDRDX(101)
      COMMON /RSOR/ RXL(101),RSS(101),RDS(101),NRSOR
      COMMON /WGEOM / XRWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1     PSIWTE(20),Y(20),PHIO(20),ZOIMED,WICR
      COMMON/WPGEOM/XRF(200),XRR(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2     SNPHI(200),CSPHI(200)
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
C
C     FORMAT STATEMENTS
C
  701 FORMAT(8F10.0)
  702 FORMAT(1H1,10X,15HRACK INPUT DATA)
  703 FORMAT(15X,13HRACK LENGTH =,F9.5,5H FEET/15X,16HMAXIMUM RADIUS =,
     1 F9.5,5H FEET/15X,57HLOCATION OF RACK NOSE RELATIVE TO LOCAL WING
     2LEADING EDGE/20X,3HX =,F9.5,5H FEET/20X,3HZ =,F9.5,5H FEET/15X,
     3 48HRACK INCIDENCE ANGLE RELATIVE TO WING ROOT CHORD/20X,4HIR =,
     4 F9.5,8H DEGREES)
  704 FORMAT(I5,F10.0)
  705 FORMAT(///15X,33HPOLYNOMIALS SPECIFYING RACK SHAPE/18X,26HX/L OF E
     1ND OF EACH SECTION/21X,7HSECTION,5X,3HX/L)
  706 FORMAT(23X,I2,3X,7F10.5)
  707 FORMAT(18X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTION/
     121X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X,
     22HC7)
  708 FORMAT(///15X,101HRACK SHAPE AS CALCULATED FROM THE INPUT POLYNOMI
     1ALS, ORIGINS OF SOURCES AND DOUBLETS REPRESENTING THE/20X,52HRACK,
     2 AND VALUES OF THE SOURCE AND DOUBLET CONSTANTS)
  709 FORMAT(///15X,21HRACK SHOCK WAVE SHAPE,6X,22HNOSE TIP SHOCK ANGLE
     1=,F7.3,8H DEGREES)
C
C     READ IN AND WRITE RACK GEOMETRY
C
      READ (5,701) RLTHC,RRMAX,XRNC,ZRN,RIC
      WRITE(6,702)
      WRITE(6,703) RLTHC,RRMAX,XRNC,ZRN,RIC
C
C     READ IN AND PRINT POLYNOMIALS DESCRIBING RACK SHAPE
C
      READ (5,704) NRPOLY
      READ (5,701) (RXEND(J),J=1,NRPOLY)
      DO 1 J=1,NRPOLY
    1 READ (5,701) (RCOEF(J,K),K=1,7)
      WRITE (6,705)
      DO 2 J=1,NRPOLY
    2 WRITE (6,706) J,RXEND(J)
      WRITE (6,707)
      DO 3 J=1,NRPOLY
    3 WRITE (6,706) J,(RCOEF(J,K),K=1,7)
C
C     LOCATE RACK IN WING COORDINATE SYSTEM
C
      YY=YPL
      IF (YY.GT.0.0) YY=-YY
      I=1
      JLE=1
      IF (YY.GE.Y(I)) GO TO 13
   12 I=I+1
      JLE=NCW*(I-1)+1
      IF (YY-Y(I)) 12,13,14
   13 XWROC=XRNC+XRF(JLE)
      YWRO=YPL
      ZWRO=ZRN
      GO TO 11
   14 JLEM=JLE-NCW
      FRAC=(YY-Y(I-1))/(Y(I)-Y(I-1))
      XWROC=XRNC+XRF(JLEM)+FRAC*(XRF(JLE)-XRF(JLEM))
      YWRO=YPL
      ZWRO=ZRN+ZRC(JLEM)+FRAC*(ZRC(JLE)-ZRC(JLEM))
   11 CONTINUE
C
C     CALCULATE LINE SOURCE AND DOUBLET DISTRIBUTIONS FOR RACK
C
      WRITE (6,708)
      RIBCR=RIC*DTOR+WICR
      ALFAR=ALFACR+RIBCR
      READ (5,704) NRSOR,RTHSHK
      NXBODY=NRSOR+1
      CALL BDYGEN(NXBODY,RRMAX,RLTHC,NRPOLY,RXEND,RCOEF,RSS,RDS,RXL,
     1ALFAR,RDRDX,RBASE,RSUMK,RSUMKD)
      NRSOR=NXBODY-1
C
C     FIND RACK SHOULDER LOCATION
C
      DO 20 N=1,NRSOR
      NN=N
      IF (RDRDX(N).LE.0.0) GO TO 21
   20 CONTINUE
   21 RXSHLD=RXL(NN)
C
C     CALCULATE RACK SHOCK WAVE SHAPE
C
      WRITE (6,709) RTHSHK
      BTNOSE=TAN((90.0-RTHSHK)*DTOR)
      CALL SHKSHP(NRSOR,RXL,RSS,BETA,RRMAX,BTNOSE,NRSHK,RXSHK,RRSHK)
      RETURN
      END


      SUBROUTINE SHAPE (X,NS,XE,C,R,DRDX)
C
C     SUBROUTINE TO CALCULATE LOCAL BODY RADIUS AND SURFACE SLOPE
C
      DIMENSION XE(7),C(7,7)
      DO 1 K=1,NS
      XL=XE(K)
      J=K
      IF (X.LE.XL) GO TO 2
    1 CONTINUE
    2 R=C(J,1)+X*C(J,5)+X*X*C(J,6)
      ARG=X*X*C(J,2)+X*C(J,3)+C(J,4)
      DRDX=C(J,5)+2.0*X*C(J,6)
      IF (ARG.LE.0.0) RETURN
      R=R+SQRT(ARG)*C(J,7)
```

Figure A-1(cc)

```
      DRDX=DRDX+(2.0*X*C(J,2)+C(J,3))/(2.0*SQRT(ARG))*C(J,7)
      RETURN
      END



      SUBROUTINE SHKSHP(NC,XS,SS,BETA,RMAX,BTNOSE,NSHK,XSHK,RSHK)
C
C     THIS SUBROUTINE CALCULATES THE SHAPE OF THE SHOCK WAVE PRODUCED
C          BY AN AXISYMMETRIC BODY AT ZERO DEGREES ANGLE OF ATTACK
C
      DIMENSION XS(1),SS(1),XSHK(1),RSHK(1)
  701 FORMAT(24X,    1HN,5X,1HX,9X,1HR,9X,1HU,9X,1HW,9X,2HML,7X,4HBETA,
     1 6X,5HDX/DR)
  702 FORMAT(20X,I5,2F10.4,5F10.5)
C
C     INITIALIZE
C
      WRITE (6,701)
      XSHK(1)=0.0
      RSHK(1)=0.0
      NSHK=1
      U=0.0
      W=0.0
      FML=SQRT(BTNOSE*BTNOSE+1.0)
      WRITE (6,702) NSHK,XSHK(1),RSHK(1),U,W,FML,BTNOSE
      DR=0.5*RMAX
      R=RMAX-DR
      X=-0.1
      DX=0.1
      BSAVE=BTNOSE
      ANGNU1=2.4495*ATAN(0.40825*BETA)
      ANGNU1=(ANGNU1-ATAN(BETA))*57.2957795
C
C     RADIAL LOCATION LOOP
C
    5 R=R+DR
      NSHK=NSHK+1
      WSAVE=0.0
      XSAVE=X
      NN=1
C
C     X LOCATION LOOP
C
   10 X=X+DX
C
C     CALCULATE SOURCE VELOCITIES
C
      U=0.0
      W=0.0
      DO 1 J=1,NC
      BETR=BETA*R
      ST=SS(J)
      ARG=(X-XS(J))/BETR
      IF (ARG.LE.1.0) GO TO 2
      ROOT=SQRT(ARG*ARG-1.0)
      ACOSH=ALOG(ARG+ROOT)
      U=U-ST*ACOSH
    1 W=W+ST*BETA*ROOT
    2 U=U+1.0
      IF (W.LT.WSAVE) GO TO 20
      WSAVE=W
      USAVE=U
      XSAVE=X
      GO TO 10
   20 IF (NN.EQ.3) GO TO 30
      X=XSAVE
      DX=0.1*DX
      NN=NN+1
      GO TO 10
C
C     CALCULATE NEW MACH NUMBER AND BETA
C
   30 TANNU=WSAVE/USAVE
      TNU=ATAN(TANNU)*57.2957795
      ANGNU2=ANGNU1-TNU
      MM=0
      IF (ANGNU2.GT.0.1) GO TO 50
   35 BETANU=BSAVE
      FNUMCH=SQRT(BETANU*BETANU+1.0)
      MM=1
      GO TO 51
   50 YNU=(ANGNU2/130.4540769)**(2.0/3.0)
      YNUSQ=YNU*YNU
      YNUC=YNU*YNUSQ
      FNUMCH=((1.0+1.3604*YNU+0.0962*YNUSQ-0.5127*YNUC)/(1.0-0.6722*YNU
     1 -0.3278*YNUSQ))*COS(TNU/57.2957795)
      IF (FNUMCH.LT.1.01) GO TO 35
      BETANU=SQRT(FNUMCH*FNUMCH-1.0)
      IF (BETANU.GT.BTNOSE) GO TO 51
      BETANU=BTNOSE
      FNUMCH=FML
   51 XSHK(NSHK)=XSHK(NSHK-1)+0.5*(BETANU+BSAVE)*(R-RSHK(NSHK-1))
      RSHK(NSHK)=R
      DXDR=(XSHK(NSHK)-XSHK(NSHK-1))/(R-RSHK(NSHK-1))
      WRITE (6,702) NSHK,XSHK(NSHK),R,USAVE,WSAVE,FNUMCH,BETANU,DXDR
      TEST=(BETA-BETANU)/BETA
      IF (TEST.LE.0.01) RETURN
      IF (MM.GT.0) GO TO 52
      IF ((BETANU-BSAVE).LT.0.025) DR=DR+DR
   52 BSAVE=BETANU
      DX=0.1
      X=XSAVE-DX
      IF (NSHK.EQ.50 ) RETURN
      GO TO 5
      END



      SUBROUTINE SMARCH (GB,VB,IA,IMAGE)
C
C  MARCHING MATRIX SOLUTION FOR SUPERSONIC FLOW.
C  SOLUTION IS SOLVED RING BY RING ALONG THE BODY ASSUMING NO INFLUENCE
C  OF PANELS ON THE BODY ON RINGS UP STREAM.
C  IF IMAGE=.TRUE., MODIFICATION TO BOUNDARY CONDITION, VB, DUE TO
C  PRESENCE OF IMAGE BODY IS CALCULATED.
C    JG = STARTING LOCATION OF GB
C    IV = STARTING LOCATION OF VB
C
      LOGICAL IMAGE
      COMMON /DIMENS/ IZDIM(79),NRING,IROW(50)
      COMMON             A(12000)
C
      DIMENSION GB(1),VB(1)
      JG = 0
      CALL IOREAD (9,IROW,NRING)
      DO 160 JB=1,NRING
      JR = IROW(JB)
      DO 150 IB=JB,NRING
      IR = IROW(IB)
      NR = IR*JR
      CALL IOREAD (9,A(IA),NR)
C
C  IF DIAGONAL BLOCK, SOLVE SIMULTANEOUS EQUATIONS
C                A * GB  =  VB
      IF (IB.NE.JB) GO TO 120
      IV = JG
      IF (IR.NE.JR) STOP 002
      CALL PAS002(A(IA),VB(JG+1),IR,1,IR,1)
      DO 100 I=1,IR
      J = I+JG
  100 GB(J) = VB(J)
      GO TO 150
C
C  FOR NON DIAGONAL BLOCKS, SOLUTION OF PRECEEDING RINGS IS KNOWN
C  EXPLICITELY.  CONTRIBUTION TO SOLUTION IS FORMED BY DIRECT
```

Figure A-1(dd)

```
C  SUBSTITUTION.  MODIFY R.H.S BY SUBTRACTING INFLUENCE FROM FORWARD
C  RINGS.    FORM PRODUCT          VB = VB - A*GB
C
120   CONTINUE
      IA1 = IA-1
      DO 140 IP=1,IR
      I = IV+IP
      IJ = IP+IA1
      SUM = 0.0
      DO 130 JP=1,JR
      J = JG+JP
      SUM = SUM+A(IJ)*GB(J)
130   IJ = IJ+IR
      VB(I) = VB(I)-SUM
140   CONTINUE
C
150   IV = IV+IR
C
C  COMPUTE INFLUENCE OF IMAGE STORE ON REMAINING PANELS
C            THE FOLLOWING ROUTINE IS CALLED ONLY IN PROGRAM II
C            WHERE THE INFLUENCE OF THE REFLECTED SHOCK IS FELT
C            REMOVE *C* IN COLUMN 1 FOR USE IN PROGRAM II
C
C     IF (IMAGE) CALL IMAGEV (GB,VB,IMAGE,JG,JR)
160   JG = JG+JR
      RETURN
      END


      SUBROUTINE SOLVE (IALP)
C
C  SOLVE FOR THE STRENGTHS OF THE BODY SOURCES
C  WHICH SATISFY THE BOUNDARY CONDITION OF TANGENTIAL FLOW AT THE
C  PANEL CONTROL POINTS.
C
C  THE PROGRAM MUST SOLVE A SYSTEM OF LINEAR EQUATIONS OF MAXIMUM
C  ORDER 1600.  FOR SUPERSONIC FLOW, THE SOLUTION TECHNIQUE CAN BE
C  DESCRIBED AS A LOWER TRIANGULAR BLOCKED L*U DECOMPOSITION METHOD.
C  THE METHOD IS PARTITIONED INTO SUBMATRICES OF MAXIMUM SIZE 32 BY 32
C  CORRESPONDING TO THE MAXIMUM NUMBER OF PANELS AROUND A RING. THE
C  MATRIX ELEMENTS IN EACH BLOCK ARE CAREFULLY CHOSEN TO REPRESENT  A
C  WELL DEFINED FEATURE OF THE ORIGINAL CONFIGURATION.  FOR BODY PANELS
C  EACH BLOCK REPRESENTS THE INFLUENCE OF ONE RING OF PANELS AROUND
C  THE BODY ON ANOTHER RING.  AFT RINGS HAVE NO INFLUENCE
C  ON FORWARD RINGS ON THE BODY.  THEREFORE ALL OFF-DIAGONAL BLOCKS
C  OF THE UPPER TRIANGLE OF THE MATRIX ARE ZERO.  THE SOLUTION IS
C  OBTAINED BY SOLVING EACH DIAGONAL BLOCK SEQUENTIALLY AND SUBSTITUTING
C  BACK INTO THE REMAINDER OF THE MATRIX .
C
C  THE ACTUAL LIMIT OF THE EQUATION SIZE IS DETERMINED FROM THREE
C  ARRAY LIMITS$ (1)  THE MAXIMUM NUMBER OF RINGS.  (2)  THE MAXIMUM
C  NUMBER OF PANELS AROUND A RING, AND (3)  THE TOTAL DIMENSION LENGTH
C  OF BLANK COMMON.
C  ---------------------------------------------------------------------
C
C  ARGUMENT LIST$
C    IALP - INDEX OF SOURCE STRENGTH VECTOR, IGB(IALP), WHICH TELLS
C           WHERE IN ARRAY - A SOLUTION IS SAVED.  THIS ALLOWS THE
C           STORAGE OF SOLUTIONS FOR MULTIPLE ANGLE OF ATTACKS.
C           IALP SHOULD BE INCREMENTED SEQUENTIALLY TO AVOID
C           OVERWRITING PREVIOUS SOLUTIONS
C
C  ---------------------------------------------------------------------
C
C  DISK FILE USAGE IN SOLVE-
C    TAPE7$  SOLVE,   REWIND
C            SOLVE,   WRITE XPT,,CP    - SAVE SOLUTION
C    TAPE9$  SOLVE,   REWIND
C            SMARCH,   READ AIC MATRIX DURING SOLUTION
C
C  BLANK COMMON USAGE IN SOLVE.
C
C     1   ENTERING ROUTINE, CONTAINS GEOMETRY
C           XPT,YPT,ZPT,THET,DELTA,AREA,XC,YC,ZC,...
C    (2) AFTER SOLUTION FOR STRENGTHS, GB
C           XPT,YPT,ZPT,THET,DELTA,AREA,XC,YC,ZC,GB(1)...GB(IALP),VB..... A.
C
      REAL NMI
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSRAD(51)
     *   ,FUSAZ(51),XU(51),PHIN(33)
      COMMON /XINLET/ NINLET,NINVEC,NTINC,AVIVO,NINNCR,XTINCT,YCPI
     *   ,XINCT,YINCT,ZINCT,XINCTE,YINCTE,ZINCTE,JINCT(25)
      COMMON /ROPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT,[illegible],IPRT(5),IOVW,XSTART,XWLE
     3   ,REFA,REFL,REFL,REFX,REFZ,CCTEST,ITMAX,BCOL,IZ1(12)
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1   ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON /DIMENS/ NX,NR,NX,NR,NINR,NXPR,NAPR,MAXNX,MAXNR,NATOT
     *   ,NBODY,NFUS,NRADX(5),NFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NACIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAPT
     *   ,IAR,IAN,I[illegible],INB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAS,NAS,NASHR,NAFLC,IAS,IC0,ISPC,IZZ,6 ,NRINS,IROW(50)
      COMMON            A 12000)
      LOGICAL SUBSON,SUPERS
      LOGICAL INLTST,INLET,OPEN
      DATA RADDEG/0.01745329267/
C
      REWIND 9
      EM=XMACH
      SUBSON = XMACH.LT.1.
      SUPERS = XMACH.GT.1.
      ALPHA=ALPHAC
      ALP=ALPHAC*RADDEG
      PHI=PHIR*RADDEG
      COSAC=COS(ALP)
      SINAC=SIN(ALP)
      CPHI=1.0
      SPHI=0.0
      SINB=0.
      SINA=SINAC
C
C  IF NON-SYMMETRIC CONFIGURATION COMPUTE COMBINED ANGLE OF ATTACK
C
      IF (IXZSYM.EQ.0) GO TO 20
      CPHI = COS(PHI)
      SPHI = SIN(PHI)
      SINB=SINAC*SPHI
      SINA=SINAC*CPHI
      BETA=ASIN(SINB)/RADDEG
      ALPHA=ASIN(SINA)/RADDEG
      IF (IPRT(5).NE.0) WRITE(6,430) ALPHAC,PHIR,ALPHA,BETA
20    CONTINUE
C              INCLUDE ONLY INLET PANELS IN STRENGTHS SOLUTION
      NTINL = NINLET*NINBLK
C
C  CALCULATE NORMAL VELOCITIES REQUIRED TO SATISFY BOUNDARY
C  CONDITIONS AT BODY CONTROL POINTS   (NBI=0)
C
      IF (IALP.LE.0) IALP=1
      IGB(IALP) = NG+(IALP-1)*NBODY+1
      IU  = IGB(IALP)+NBODY
      IV  = IU +NBODY
      IW  = IV +NBODY
      IVA = IW +NBODY
      IVB = IU
      IAN = IVA
      DO 30 J=1,NBODY
      JM1 = J-1
      JDEL = IDEL+JM1
      JTH  = ITH+JM1
      JVB  = IVB+JM1
      JV   = IV +JM1
      DELTA = A(JDEL)
      THET  = A(JTH)
      NFI=COSAC*SIN(DELTA)-COS(DELTA)*(SINA*COS(THET)+SINB*SIN(THET))
```

Figure A-1(ee)

```
C
C  IF INLET, MULTIPLY R.C. BY INLET SPILLAGE RATIO
C
      FINLET = 1.0
      IF (.NOT.INLTST(J,OPEN)) GO TO 25
      IF (       OPEN) FINLET=0.0
      IF (.NOT.OPEN) FINLET=1.0
      NBI = NBI*FINLET
25    A(JVB) = NBI
      A(JV)  = NBI
30    CONTINUE
C
C ----------------------------------------------------------------------
C  SOLVE MATRIX EQUATIONS -
C  IN SUPERSONIC FLOW MATRIX IS SOLVED BY MARCHING RING BY RING DOWN
C  BODY.  AFT RINGS ARE ASSUMED NOT TO INFLUENCE FORWARD PANELS.
C
      IGBA = IGB(IALP)
C                     GR        VB
      CALL SMARCH (A(IGBA),A(IVB),IAN,.FALSE.)
C
C  COMPUTE NORMAL VELOCITIES AT CONTROL POINTS
C
      IF (IPRT(5).LE.0) GO TO 210
      WRITE (6,350) EM,ALPHAC,PHIR
      WRITE (6,400)
      DO 200 J=1,NBODY
      JM1 = J-1
      JV  = IV +JM1
      JGB = IGB(IALP)+JM1
      VB = A(JV)
      GR = A(JGB)
      WRITE (6,420) J,GR,VB
200   CONTINUE
C
C  SAVE SOLUTION WITH GEOMETRY ON TAPE7 --------------------------------
C
210   REWIND 7
      NTAP7 = IU-1
      CALL IOWRIT (7,A,NTAP7)
      RETURN
350   FORMAT (///10X,16HVELOCITY ON BODY,84X,12H** SOLVE  **,
     *  //5X,5HMACH=,F6.3,3X,6HALPHA=,F7.3,3X,5HPHIR=,F7.3//)
390   FORMAT (/9H CPSTAG =,F10.5,3X,8HCPCRIT =,F10.5,3X,7HCPVAC =,F10.5)
400   FORMAT (8X,5HPANEL,3X,6HSOURCE, 5X,9HNORMAL BC
     2   ,/8X,3HNO.,4X,8HSTRENGTH,4X,8HVELOCITY)
420   FORMAT (I12,F10.5,G15.5)
430   FORMAT(//10X,25HCOMBINED ANGLES OF ATTACK,75X,12H** SOLVE **
     *    ,/5X,7HALPHAC=,F10.4,5X,7H  PHIR=,F10.4
     *    ,/5X,7H ALPHA=,F10.4,5X,7H  BETA=,F10.4)
      END


      SUBROUTINE SORPAN(UPM,VPM,WPM)
C
C  COMPUTE THE THREE COMPONENTS OF VELOCITY INDUCED AT A SPECIFIED
C  CONTROL POINT BY A CONSTANT SOURCE DISTRIBUTION ON A
C  QUADRILATERAL PANEL HAVING LONGITUDINAL TAPER AND INCLINED AT AN
C  ANGLE DELTA TO THE FREE STREAM DIRECTION
C  ******************** SUPERSONIC VERSION ****************************
C
      COMMON /BODCOM/ AMACH,TAND,CX,XC(4),YC(4),ZC(4)
     1   ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,J
      DIMENSION SX(4),SM(4),DX(4),DY(4),DZ(4),D(4),E(4),F(4),G(4),
     1 H(4),XPM(4),YMX(4),ZAX(4),AYM(4),RPM2(4)
C
      REAL NUM
      LOGICAL SUBSON,SUPERS
      DATA EPS,PI/1.E-5,3.14195265/
      IF (SUBSON) GO TO 220
C
      EP2=EPS*EPS
      BT2 = BETAL*BETAL
      BTA = BETAL
      TA=1.-BT2*TAND*TAND
      IF (TA.LT.0.) GO TO 200
      TAD=1./TA
      SM(1)=(YC(2)-YC(1))/CX
      SM(3)=(YC(4)-YC(3))/CX
      SM(2)=SM(1)
      SM(4)=SM(3)
      DO 190 I=1,4
      ZC(I)=ZJ-TAND*(XJ-XC(I))
      SSM=SIGN(1.,SM(I))
      TAM=TA-BT2*SM(I)*SM(I)
      IF (ABS(TAM).LE.EPS) TAM=0.
      SAM=SQRT(ABS(TAM))
      SAMD=1./SAM
      CPM=CX*SAM
      DX(I)=XI-XC(I)
      DY(I)=YI-YC(I)
      DZ(I)=ZI-ZC(I)
      IF (ABS(DX(I)).LE.EPS) DX(I)=0.
      IF (ABS(DY(I)).LE.EPS) DY(I)=0.
      IF (ABS(DZ(I)).LE.EPS) DZ(I)=0.
      RPM2(I)=0.
      D2=DX(I)*DX(I)-BT2*(DY(I)*DY(I)+DZ(I)*DZ(I))
      D(I)=0.
C               CHECK FOR POINT LAYING WITHIN MACH CONE
      DXZ=DX(I)-BTA*ABS(DZ(I))
      IF (DXZ.LT.0.) GO TO 170
      IF (D2.GT.0.) D(I)=SQRT(D2)
      XPM(I)=DX(I)-BT2*(SM(I)*DY(I)+TAND*DZ(I))
      YMX(I)=DY(I)-SM(I)*DX(I)
      ZAX(I)=DZ(I)-TAND*DX(I)
      AYM(I)=TAND*DY(I)-SM(I)*DZ(I)
      IF (ABS(XPM(I)).LE.EPS) XPM(I)=0.
      IF (ABS(YMX(I)).LE.EPS) YMX(I)=0.
      IF (ABS(ZAX(I)).LE.EPS) ZAX(I)=0.
      IF (ABS(AYM(I)).LE.EPS) AYM(I)=0.
      IF (I.LE.2) RPM2(1)=YMX(1)**2+ZAX(1)**2-BT2*AYM(1)**2
      RPM2(2)=RPM2(1)
      IF (I.GT.2) RPM2(3)=YMX(3)**2+ZAX(3)**2-BT2*AYM(3)**2
      RPM2(4)=RPM2(3)
      IF (ABS(RPM2(I)).LE.EP2) RPM2(I)=0.
      RPM=SQRT(ABS(RPM2(I)))
      IF (RPM.LE.EPS) RPM=0.
      DPM=SAM*D(I)
      F(I)=0.
      DNOM=-DX(I)*YMX(I)+BT2*DZ(I)*AYM(I)
      FNUM=D(I)*ZAX(I)
      IF (FNUM.EQ.0..AND.DNOM.EQ.0.) GO TO 10
      IF (D(I).EQ.0..OR.ZAX(I).EQ.0.) FNUM=0.
      F(I)=ATAN2(FNUM,DNOM)
      IF (D(I).EQ.0.) F(I)=F(I)*SIGN(1.,ZAX(I))
10    IF (TAM) 100,90,20
20    IF (D(I).EQ.0.) GO TO 70
      IF (RPM.LE.EPS) GO TO 40
      NUM=XPM(I)+DPM
      G(I)=ALOG(NUM/(BTA*RPM))*SAMD
      GO TO 160
40    SX(I)=SIGN(1.,XPM(I))
      IF (I.EQ.1.AND.XPM(1).LT.CPM) GO TO 130
      IF (I.EQ.3.AND.XPM(3).LT.CPM) GO TO 140
50    IF (I.EQ.2) SGN12=SX(1)*SX(2)
      IF (I.EQ.4) SGN34=SX(3)*SX(4)
      IF (XPM(I)) 60,70,80
60    IF (I.EQ.2.AND.SGN12.LT.0.) GO TO 130
      IF (I.EQ.4.AND.SGN34.LT.0.) GO TO 140
      G(I)=-ALOG(ABS(XPM(I)))*SAMD
      GO TO 160
70    G(I)=0.
      GO TO 160
80    G(I)=ALOG(XPM(I))*SAMD
      GO TO 160
90    G(I)=0.
```

Figure A-1(ff)

```
      IF (XPM(I).GT.BTA*RPM) G(I)=D(I)/XPM(I)
      GO TO 160
100   G(I)=0.
      ARG=SIGN(1.,XPM(I))
      IF (RPM.NE.0.) ARG=XPM(I)/(BTA*RPM)
      IF (ARG.GE.1.) GO TO 160
      IF (ARG.LE.-1.) GO TO 110
      IF (D(I).GT.0.) G(I)=ACOS(ARG)*SAMD
      GO TO 160
110   TRM1=(SM(I)*DY(I)+TAND*DZ(I)+ABS(AYM(I))*SAM)/(TAND*TAND+SM(I)**2)
      IF (DX(I).GT.TRM1) GO TO 120
      F(I)=0.
      IF (SSM.GT.0.) F(I)=PI*SIGN(1.,ZAX(I))
      GO TO 160
120   IF (SSM*YMX(I).GE.0.) GO TO 160
      G(I)=PI*SAMD
      GO TO 160
130   G(I)=500.*SAMD
      GO TO 160
140   G(3)=500.*SAMD
160   H(I)=0.
      HARG=-BTA*DY(I)
      IF (D(I).EQ.0..AND.HARG.EQ.0.) GO TO 180
      H(I)=BTA*ATAN2(D(I),HARG)
      GO TO 180
C
C  SET CORNER EQUAL TO ZERO. POINT AHEAD OF MACH CONE
C
170   F(I)=0.
      G(I)=0.
      H(I)=0.
      AYM(I)=0.
      YMX(I)=0.
      ZAX(I)=0.
      XPM(I)=0.
      DPM=0.
      RPM=0.
      RPM2(2)=RPM2(1)
      RPM2(4)=RPM2(3)
180   E(I)=H(I)-BT2*SM(I)*G(I)
190   CONTINUE
C
C  END LOOP OVER FOUR CORNERS.  SUM INFLUENCE OF CORNERS TO CALCULATE
C  U,V,W IN PANEL COORDINATE SYSTEM.
C
      TAD=1./TA
      E14=(E(1)-E(2)-E(3)+E(4))*TAD
      F14=(F(1)-F(2)-F(3)+F(4))*TAD
      G14=G(1)-G(2)-G(3)+G(4)
      R4PI=1./(2.*PI)
      UPM=-R4PI*(E14/BT2+TAND*F14)
      VPM=-R4PI*G14
      WPM=R4PI*(F14+TAND*E14)
      RETURN
C             ERROR CONDITION  - PANEL AHEAD OF MACH CONE
200   WRITE (6,210) J
      WRITE(6,888) AMACH,TAND,CX,XC,YC,ZC,XI,YI,ZI,XJ,ZJ,BETAO,BETAL
888   FORMAT(10G12.5)
      STOP 210
210   FORMAT (1H0,47HERROR - BODY PANEL SLOPE EXCEEDS MACH ANGLE, J=,I4,
     *   63X,12H** SORPAN **)
220   WRITE(6,230)
      STOP 230
230   FORMAT (//10(2H *),27H MACH NUMBER .LT.1,  SORPAN)
      END


      SUBROUTINE SOURCE(TX)
C
C SUBROUTINE TO CALCULATE THE STRENGTH OF A LINEAR LINE SOURCE OF
C UNIT SLOPE WITH ORIGIN AT TX(J).
C
      COMMON /FLOW/DUM1(5),BETA,DUM2(2,
      COMMON/SRCE/XFIELD,RFIELD,BSQ,X2,      U,V,VT
C
      X1=XFIELD-TX
      BR=BETA*RFIELD
      IF(X1.LE.BR) GO TO 10
      XL=X1/BR
      D23= SQRT(XL*XL-1.)
      U=-ALOG(XL+D23)
      V=BETA*D23
      RETURN
C
C FIELD POINT IS AHEAD OF MACH CONE FROM SOURCE ORIGIN.
C
   10 U=0.
      V=0.
      RETURN
      END


      SUBROUTINE STORIO
C
      SUBROUTINE TO INPUT AND OUTPUT STORE DATA
          LINE SOURCE AND DOUBLET DISTRIBUTIONS ARE CALCULATED
          USING SUBROUTINE HOYGEN
C
      DIMENSION XSNC(7),YSN(7),ZSN(7),SIC(7),SPHIR(7)
      DIMENSION SXNDJ(7),SCOFJ(7,7),SHKANG(7)
C
      COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7),
     1 SSUMK(7),SSUMKD(7)
      COMMON /PGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSRAD(51)
     * ,FUSAZ(51),XJ(51),PHIK(33)
      COMMON /POPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BOOL,IZ1(12)
      COMMON /RPGEOM/ SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1      SANGW
      COMMON /CONFIG/ NFU,NPY,NSTRS,LVSWP,NRACK
      COMMON /CONSTS/ PI,PI2,DTOR,RTOD,FOURPI
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ,FMCHSQ
      COMMON /INDEX /NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON /PARAM / XMACH,ALPHA,BETAY,ALPHAC,PHIR,EM
     1   ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON /SSHOCK/ NSSHK(7),SXSHK(50,7),SRSHK(50,7),SDHDX(101,7)
      COMMON /SSOR  / SXL(101,7),SSS(100,7),SDS(100,7),NSSOR(7)
      COMMON /STGEOM/ SRMAX(7),SLTHC(7),XWSOC(7),YWSO(7),ZWSO(7),
     1       NSPOLY(7),SXEND(7,7),SCOEF(7,7,7),XBSO(7),YBSO(7),ZBSO(7),
     2       SIBCR(7),NUMSTR(7),SSIBCR(7),CSIBCR(7),SPHIRR(7),NSHAPE(7)
     3       ,NSHPT,MSHAPE(7)
      COMMON /WGEOM / XPWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1       PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR
      COMMON /WPGEOM/ XRF(200),XRR(200),XLF(200),XLR(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2      SNPHI(200),CSPHI(200)
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
C
701 FORMAT (2I5,7F10.0)
702 FORMAT(1H1,10X,16HSTORE INPUT DATA)
703 FORMAT(/40X,36HMAXIMUM   STORE LOCATION RELATIVE TO,6X,9HINCIDENCE
   1,5X,4HROLL/15X,20HSTORE SHAPE   LENGTH,5X,6HRADIUS,4X
   2,29HLOCAL WING CHORD LEADING EDGE,5X,5HANGLE,6X,5HANGLE
   3/17X,2HNO,4X,2HNO,6X,2HFT,9X,2HFT,8X,5HX, FT,
   46X,5HY, FT,6X,5HZ, FT,6X,3HDEG,8X,3HDEG)
704 FORMAT(15X,2I5,7F11.5)
707 FORMAT(///5X,48H***** SHAPE POLYNOMIALS NOT INPUT FOR ALL STORES, 
   1        57X,12H** STORIO **)
712 FORMAT (8F10.5)
713 FORMAT (///15X,37HPOLYNOMIALS SPECIFYING STORE SHAPE NO.I3/
   1        18X,26HX/L OF END OF EACH SECTION/21X,7HSECTION,5X,3HX/L)
```

Figure A-1(gg)

```
714 FORMAT (23X,I2,3X,7F10.5)
715 FORMAT(15X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTION/
   121X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X,
   22HC7)
716 FORMAT(///15X,103HSTORE SHAPES AS CALCULATED FROM THE INPUT POLYNO
   1MIALS, ORIGINS OF SOURCES AND DOUBLETS REPRESENTING THE/20X,
   2 54HSTORES, AND VALUES OF THE SOURCE AND DOUBLET CONSTANTS)
717 FORMAT(///15X,23HSTORE SHOCK WAVE SHAPES)
718 FORMAT(/18X,8HSTORE NO.I3)
719 FORMAT(/18X,8HSTORE NO.I3,6X,22HNOSE TIP SHOCK ANGLE =,F7.3,
   1 8H DEGREES)
720 FORMAT (//15X,22HNOSE TIP SHOCK ANGLE =,F10.3)
721 FORMAT(///15X,24HDATA FOR STORE SHAPE NO.,I5
   *   ///5X,49HELLIPTIC STORE IS MODELLED USING WOODWARD SOURCE
   *   ,12HPANEL OPTION)
734 FORMAT(/15X,25H   STORE SHOCK WAVE SHAPE)
    REWIND 11
C
C
C     READ IN AND WRITE STORE GEOMETRY
C
C
      WRITE (6,702)
      WRITE (6,703)
      DO 2 J=1,NSTRS
      READ (5,701) NUMSTR(J),NSHAPE(J),SLTHC(J),SRMAX(J),XSNC(J),
    1      YSN(J),ZSN(J),SIC(J),SPHIR(J)
      WRITE (6,706) NUMSTR(J),NSHAPE(J),SLTHC(J),SRMAX(J),XSNC(J),
    1      YSN(J),ZSN(J),SIC(J),SPHIR(J)
    2 CONTINUE
      NCOUNT=0
C
C     READ IN AND PRINT DATA DESCRIBING STORE SHAPE
C
      READ (5,701) NSHPT
      DO 70 J=1,NSHPT
      READ (5,701) MSHAPE(J),MSOR,STHSHK
      IF (MSHAPE(J).GT.50) GO TO 53
C
C     CIRCULAR STORE SHAPE DATA
C
      WRITE (6,713) MSHAPE(J)
      READ(5,7   NSPOLJ
      READ(5,     SXNDJ(K),K=1,NSPOLJ)
      DO 50 K=     POLJ
   50 READ (5,    ) (SCOFJ(K,L),L=1,7)
      DO 51 K=1,NSPOLJ
   51 WRITE (6,714) K,SXNDJ(K)
      WRITE (6,715)
      DO 52 K=1,NSPOLJ
   52 WRITE (6,714) K,(SCOFJ(K,L),L=1,7)
      GO TO 60
C
C     READ ADDITIONAL GEOMETRY INPUT REQUIRED FOR NONCIRCULAR STORE
C     SAVE STORE CONFIGURATION GEOMETRY ON TAPE 11
C
   53 CONTINUE
      WRITE(6,721) MSHAPE(J)
      IPRES = 1
      CALL GEOM
      CALL FRSTRT (11,1,LASTA)
C
C     SET DATA FOR STORE K EQUAL TO THIS INPUT DATA IF SHAPE NUMBERS ARE
C          THE SAME
C
   60 DO 68 K=1,NSTRS
      IF (MSHAPE(J).NE.NSHAPE(K)) GO TO 68
      SHKANG(K)=STHSHK
      IF (NSHAPE(K).GT.50) GO TO 66
      NSSOR(K)=MSOR
      NSPOLY(K)=NSPOLJ
      DO 62 I=1,NSPOLJ
      SXEND(I,K)=SXNDJ(I)
      DO 62 IC=1,7
   62 SCOEF(I,IC,K)=SCOFJ(I,IC)
   66 NCOUNT=NCOUNT+1
   68 CONTINUE
   70 CONTINUE
C
C     WRITE ERROR MSG IF DATA IS MISSING FOR SOME STORE
C
      IF (NCOUNT .EQ. NSTRS) GO TO 10
      WRITE (6,707)
      STOP 707
C
C     LOCATE STORES IN WING COORDINATE SYSTEM
C
   10 CONTINUE
      DO 11 J=1,NSTRS
      YY=YSN(J)
      IF (YY .GT. 0.0) YY=-YY
      I=1
      JLE=1
      IF(YY.GE.Y(1)) GO TO 13
   12 I=I+1
      JLE=NCW*(I-1) +1
      IF(YY-Y(I)) 12,13,14
   13 X=SOC(J)=XSNC(J)+XRF(JLE)
      Y=SO(J)=YSN(J)
      Z=SO(J)=ZSN(J)
      GO TO 11
   14 JLEM=JLE-NCW
      FRAC=(YY-Y(I-1))/(Y(I)-Y(I-1))
      X=SOC(J)=XSNC(J)+XRF(JLEM)+FRAC*(XRF(JLE)-XRF(JLEM))
      Y=SO(J)=YSN(J)
      Z=SO(J)=ZSN(J)+ZRC(JLEM)+FRAC*(ZRC(JLE)-ZRC(JLEM))
   11 CONTINUE
C
C     CALCULATE SOURCE AND DOUBLET DISTRIBUTIONS FOR ALL STORES
C
      REWIND 11
      DO 130 ISHP=1,NSHPT
      IF   (MSHAPE(ISHP).LE.50) GO TO 100
      CALL FRSTRT (11,2,LASTA)
      EM = 0.0
  100 CONTINUE
      ISTR = 0
C
      DO 120 K=1,NSTRS
      IF (NSHAPE(K).NE.MSHAPE(ISHP)) GO TO 120
      ISTR = ISTR+1
      SIRCR(K)=SIC(K)*DTOR+WICR
      SPHIRR(K) = SPHIR(K)*DTOR
      ALFAS=ALFACR+SIRCR(K)
      WRITE (6,718) NUMSTR(K)
      IF (NSHAPE(K).GT.50) GO TO 110
C
C             USE LINE SOURCE/DOUBLET METHOD
      WRITE (6,716)
      NXBODY=NSSOR(K)+1
      CALL BOYGEN(NXBODY,SRMAX(K),SLTHC(K),NSPOLY(K),SXENU(1,K)
     1  ,SCOEF(1,1,K),SSS(1,K),SDS(1,K),SXL(1,K),ALFAS,SORDX(1,K)
     2  ,SRASE(K),SSUMK(K),SSUMKD(K))
      MSOR=NXBODY-1
      NSSOR(K)=MSOR
C
C     FIND STORE SHOULDER LOCATION
C
      DO 105 N=1,MSOR
      IF (SORDX(N,K).LE.0.0) GO TO 106
  105 CONTINUE
      N = MSOR
  106 SXSHLD(K) = SXL(N,K)
C
C     CALCULATE SHOCK WAVE SHAPES FOR STORE-K
C
      WRITE (6,717)
      STHSHK=SHKANG(K)
```

Figure A-1(hh)

```
      BTNOSF=TAN((90.0-STHSHK)*DTOR)
      WRITE (6,719) NUMSTR(K),STHSHK
      CALL SHKSHP(NSSOR(K),SXL(1,K),SSS(1,K),BETA,SRMAX(K),BTNOSE,
     1   MSSHK,SRSHK(1,K),SXSHK(1,K))
      NSSHK(K)=MSSHK
      GO TO 120
C
C              USE SOURCE PANEL METHOD FOR STORE SOLUTION
C              RESTORE STORE CONFIGURATION GEOMETRY
C              GENERATE STORE SHOCK SHAPE AT ALPHA=PHIR=0.
C
C
110   XMACH = FMACH
      CALL VELCMP(1)
      IF (ISTR.GT.1) GO TO 115
      ALPHAC = 0.0
      PHIR   = 0.0
      CALL SOLVE (ISTR)
C
C  ELLIPTIC STORE IS ASSUMED TO BE SYMMETRIC RIGHT-TO-LEFT , TOP-TO-BOT
C
      WRITE(6,734)
      WRITE (6,720) STHSHK
      SRMAX(K) = 0.0
      N = 0
      DO 112 J=1,NFUS
      NFUSX = NFORX(J)
      DO 112 I=1,NFUSX
      N = N+1
112   SRMAX(K) = AMAX1(SRMAX(K),FUSAZ(N),FUSRAD(N))
      CALL BSHOCK (SRMAX(K),90.0,SHKANG(K))
C
C              COMPUTE SOURCE STRENGTHS FOR STORE AT GIVEN ALFAS AND PHIR
C
115   ALPHAC = ALFAS*RTOD
      PHIR   = SPHIR(K)
      CALL SOLVE (ISTR)
C
C     SAVE GEOMETRY, STRENGTHS AND AERODYNAMIC MATRICES ON TAPE10
C
      CALL FRSTRT (10,3,LASTA)
120   CONTINUE
130   CONTINUE
      RETURN
      END


      SUBROUTINE SWNGIN
C
C     SUBROUTINE TO INPUT WING CONSTANT U-VELOCITY PANEL
C       DATA AND TWIST/CAMBER DISTRIBUTION IF ANY
C
      LOGICAL ZDIHED
      COMMON/CAMBER/ALPHAL(200)
      COMMON/FSGEOM/FRMAX,SSPAN,FLTHC,BODYPL
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,NIP,N2,N2P,NPTOT
      COMMON /WGEOM / XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1      PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR
 701  FORMAT(10I5)
 706  FORMAT(8F10.0)
7020  FORMAT(15X,25HLENGTH OF WING ROOT CHORD/25X,5HCRW =,F10.5,5H FEET/
     115X,13HWING SEMISPAN/25X,7HSSPAN =,F10.5,5H FEET)
7040  FORMAT(I5,4F10.5)
7050  FORMAT(/14X,I4,  56H U-VELOCITY PANELS ARE TO BE LAID OUT ON EACH
     1WING PANEL/20X,I3,20H CHORDWISE ROWS WITH,I3,12H IN EACH ROW/)
7060  FORMAT(15X,59HSPANWISE LOCATIONS OF PANEL SIDE EDGES AND SWEEP ANG
     1LES AND/20X,43HDIHEDRAL ANGLE OF WING SECTION TO THE RIGHT.
     2  //26X,39HSPANWISE   LE SWEEP  TE SWEEP  DIHEDRAL/
     3  20X,1HI,5X,8HLOCATION/28X,4HFEET,5X,7HDEGREES,3X,7HDEGREES,3X,
     4   7HDEGREES)
7070  FORMAT(18X,I3,4X,4F10.5)
C
      READ(5,706) CRW,SSPAN
      WRITE(6,7020) CRW,SSPAN
      READ(5,701) NCW,MSW
      NPANLS=NCW*MSW
      MSWP=MSW+1
      DO 11 J=1,MSWP
   11 READ(5,7040) I,Y(I),PSIWLE(I),PSIWTE(I),PHID(I)
      WRITE(6,7050) NPANLS,MSW,NCW
      WRITE(6,7060)
      DO 13 J=1,MSWP
   13 WRITE(6,7070) J,Y(J),PSIWLE(J),PSIWTE(J),PHID(J)
C
C     SET LOGICAL VARIABLE
C
      ZDIHED=.TRUE.
      DO 14 J=1,MSWP
   14 IF(PHID(J).NE.0.0)ZDIHED=.FALSE.
C
C     INPUT TWIST AND CAMBER DISTRIBUTION, IF ANY.
C
      DO 15 J=1,NPANLS
   15 ALPHAL(J)=0.0
      READ (5,701) NTAC,NUNI
      IF (NTAC .EQ. 0) GO TO 19
      IF (NUNI .NE. 0) GO TO 17
      MN=0
      DO 16 JNW=1,NPANLS,NCW
      MN=MN+NCW
   16 READ (5,706) (ALPHAL(J),J=JNW,MN)
      GO TO 19
   17 READ (5,706) (ALPHAL(J),J=1,NCW)
      DO 18 J=2,MSW
      JJ=(J-1)*NCW
      DO 18 K=1,NCW
      KK=JJ+K
   18 ALPHAL(KK)=ALPHAL(K)
   19 CONTINUE
      RETURN
      END


      SUBROUTINE THKLYT
C
C     SUBROUTINE TO LAY OUT WING AND PYLON THICKNESS PANELS
C
      COMMON /CONFIG/NFU,NPY,NSTRS,LVSWP,NRACK
      COMMON/CONSTS/PI,PI2,DTOR,RTOD,FOURPI
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,NIP,N2,N2P,NPTOT
      COMMON/PYGEOM/Z(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
     1   SLLE,PSLPDF,CENTER,ZPL,LVSPP
      COMMON/THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRRT(400),
     1   XLFT(400),XLRT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2   THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3   YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4   PSPSTE(20),SNPHS(400),CSPHS(400)
      COMMON /WGEOM / XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1      PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR
      COMMON/WPGEOM/XRF(200),XRR(200),XLF(200),XLR(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2   SNPHI(200),CSPHI(200)
C
C     LAY OUT THICKNESS PANELS ON LEFT WING PANEL
C
      SLPDIF=SLPWLE-SLPWTE
      CSIDEP=CRW
      WLEX=0.0
      ZLER=0.0
      ANCW=NCWS
C
C     LOOP OVER CHORDWISE ROWS OF THICKNESS PANELS
C                  I IS THE SPAN INDEX
```

Figure A-1(ii)

```
C
      MS1=MSWS+1
      DO 140 I=2,MS1
      IM=I-1
      DY=YS(I)-YS(IM)
      IF(LVSWP.EQ.0) GO TO 40
      SLPWLE=TAN(PSWSLE(I)*DTOR)
      SLPWTE=TAN(PSWSTE(I)*DTOR)
      SLPDIF=SLPWLE-SLPWTE
   40 CSIDE=CSIDEP+DY*SLPDIF
      IF(I.EQ.2) GO TO 45
      JLE=(I-3)*NCWS+1
      WLEX=XLFT(JLE)
      ZLER=ZLCT(JLE)
   45 CONTINUE
      CSP=COS(PHIS(I)*DTOR)
      SNP=SIN(PHIS(I)*DTOR)
C
C     LOOP OVER THICKNESS PANELS IN THIS CHORDWISE ROW
C
      DO 130 K=1,NCWS
      J=(I-2)*NCWS+K
      AK=K
      AKM=K-1
      SLLET(J)=SLPWLE-AKM*SLPDIF/ANCW
      SLTET(J)=SLPWLE-AK*SLPDIF/ANCW
      CSPHS(J)=CSP
      SNPHS(J)=SNP
C
C     CORNER POINT COORDINATES
C
      XRFT(J)=-AKM*CSIDEP/ANCW+WLEX
      XRBT(J)=XRFT(J)-CSIDEP/ANCW
      XLFT(J)=-AKM*CSIDE/ANCW+DY*SLPWLE+WLEX
      XLBT(J)=XLFT(J)-CSIDE/ANCW
      YRCT(J)=YS(IM)
      YLCT(J)=YS(I)
      ZRCT(J)=ZLER
      ZLCT(J)=ZLER+(SNP/CSP)*DY
  130 CONTINUE
      CSIDEP=CSIDE
  140 CONTINUE
C
C     LAY OUT THICKNESS PANELS ON PYLON IF PRESENT
C
      IF(NPY.EQ.0) RETURN
      CSIDEP=CRP
      PLEX=XPLE
      INDX=(IP-2)*NCW+1
      IF(IP.GT.1) PLEX=XPLE+XLF(INDX)
      ANCP=NCPS
C
C     LOOP OVER CHORDWISE ROWS OF PYLON THICKNESS PANELS
C
      MSPP=MSPS+1
      DO 300 I=2,MSPP
      IF(LVSPP.EQ.0) GO TO 150
      SLL =TAN(PSPSLE(I)*DTOR)
      SLTE=TAN(PSPSTE(I)*DTOR)
      PSLPDF=SLLE-SLTE
  150 IM=I-1
      DZ=ZS(IM)-ZS(I)
      CSIDE=CSIDEP+DZ*PSLPDF
      IF(I.EQ.2) GO TO 201
      JLE=(I-3)*NCPS+1+MS
      PLEX=XLFT(JLE)
  201 CONTINUE
C
C     LOOP OVER PYLON THICKNESS PANELS IN THIS CHORDWISE TOW
C
      DO 250 K=1,NCPS
      J=(I-2)*NCPS+K+MS
      AKM=K-1
      AK=K
C
C     THICKNESS PANEL LEADING AND TRAILING EDGE SLOPES
C
      SLLET(J)=SLLE-AKM*PSLPDF/ANCP
      SLTET(J)=SLLE-AK*PSLPDF/ANCP
C
C     CORNER POINT COORDINATES
C
      XRFT(J)=-AKM*CSIDEP/ANCP+PLEX
      XRBT(J)=XRFT(J)-CSIDEP/ANCP
      XLFT(J)=-AKM*CSIDE/ANCP+DZ*SLLE+PLEX
      XLBT(J)=XLFT(J)-CSIDE/ANCP
      YLCT(J)=YPL
      YRCT(J)=YPL
      ZRCT(J)=ZS(IM)+ZPL
      ZLCT(J)=ZS(I)+ZPL
  250 CONTINUE
      CSIDEP=CSIDE
  300 CONTINUE
C
      RETURN
      END



      SUBROUTINE THKOUT
C
C     SUBROUTINE TO OUTPUT THICKNESS SURFACE SLOPES
C
      COMMON /CONFIG/NFU,NPY,NSTRS,LVSWP,NRACK
      COMMON/CONSTS/PI,PI2,DTOR,RTOD,FOURPI
      COMMON/THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     1   XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2   THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3   YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4   PSPSTE(20),SNPHS(400),CSPHS(400)
C
  720 FORMAT(1H1,20X,56HINPUT VALUES OF THE LOCAL SURFACE SLOPE OF THE T
     1HICKNESS/22X52HDISTRIBUTION. FOR EACH CHORDWISE ROW THE FIRST VALU
     2E/27X,41HIS FOR THE PANEL NEAREST THE LEADING EDGE)
  721 FORMAT(/10X,19HWING THICKNESS DATA)
  722 FORMAT(17X,3HROW,37X,6HSLOPES)
  723 FORMAT(15X,I5,8F10.5,9(/20X,8F10.5))
  724 FORMAT(/10X,20HPYLON THICKNESS DATA)
C
C     OUTPUT WING SLOPES
C
      WRITE (6,720)
      WRITE (6,721)
      WRITE (6,722)
      MN=0
      DO 32 JNW=1,MS,NCWS
      MN=MN+NCWS
      I=MN/NCWS
   32 WRITE (6,723) I,(THETAL(J),J=JNW,MN)
C
C     OUTPUT PYLON SLOPES, IF PRESENT
C
      IF (NPY .EQ. 0) GO TO 38
      WRITE (6,724)
      WRITE (6,722)
      MN=0
      DO 35 JNW=1,MPS,NCPS
      MN=MN+NCPS
      I=MN/NCPS
   35 WRITE (6,723) I,(THETPL(J),J=JNW,MN)
C
C     DIVIDE INPUT SLOPES BY PI AND STORE IN DZDX ARRAY
C
      DO 36 I=1,MPS
      IP=MS+I
   36 DZDX(IP)=THETPL(I)/PI
   38 CONTINUE
```

Figure A-1(jj)

```
      DO 37 I=1,MS
   37 DZDX(I)=THETAL(I)/PI
      RETURN
      END




      SUBROUTINE VELBD1(XX,YY,ZZ)
C
C     CALCULATES THE INFLUENCE COEFFICIENTS DUE TO A HODY INTERFERENCE
C     AT THE CONTROL POINT OF A CONSTANT U VELOCITY PANEL
C     INFLUENCE COEFFICIENTS UP,VP,WP ARE RETURNED IN THE WING
C     REFERENCE FRAME
C
      COMMON/BPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1      SANGW
      COMMON/ICVEL/UP,VP,WP,II,IF,          DELTP(200)
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NHIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON/WPGEOM/XRF(200),XRR(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2      SNPHI(200),CSPHI(200)
      COMMON /VELARG/X,Y,Z,U,V,W,EM,TLRNC,TIPY,PYPNL
C
      LOGICAL       PYPNL,FELTA,FELTAI,FELTB,FELTBI,FELT
C
C     THERE ARE TWO COORDINATE SYSTEMS IN USE IN THIS SUBROUTINE --
C
C     SYSTEM 1 - WING COORDINATE SYSTEM
C               X,U POSITIVE FORWARD
C               Y,V POSITIVE TO RIGHT
C               Z,W POSITIVE DOWN
C
C     SYSTEM 2 - ORIGIN AT VERTEX OF INFLUENCING SEMI-INFINITE TRIANGLE
C     (REQUIRED BY SUBROUTINE VELO1)
C               X,U POSITIVE BACK
C               Y,V POSITIVE CLOCKWISE LOOKING FORWARD, IN PLANE OF
C                  INFLUENCING PANEL
C               Z,W POSITIVE OUTWARD NORMAL TO INFLUENCING PANEL
C
C     INFLUENCE COEFFICIENTS UP,VP,WP ARE IN SYSTEM 1
C
      PYPNL=.FALSE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      YI=YY
      ZI=ZZ
      EM=0.0
C
C     I IS THE INDEX OF THE INFLUENCING PANEL
C
      I=II
      K=I-N2
      CS=CST2(K)
      SN=SNT2(K)
      FELT=.FALSE.
      TU=0.0
      TV=0.0
      TW=0.0
      IF(XI.GE.XLF(I)) GO TO 40
      IF((THTI(K).GT.90..AND.THTI(K).LE.180.)) GO TO 15
C
C     ******************* CORNER 1 ***********************************
C
C
C     EQUATIONS FOR PANELS IN LOWER LEFT QUADRANT
C
C
C     X IS IN SYSTEM 2
C
      X=XLF(I)-XI
```

```
      YW=YI-YLC(I)
      ZW=ZI-ZLC(I)
C
C     TRANSFORM COORDINATES YW,ZW IN SYSTEM 1 TO Y,Z IN
C     SYSTEM 2 FOR PANEL NUMBER I
C
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 11
C
C     CHANGE SIGN OF U TO RETURN IT TO SYSTEM 1
C
      U=-U
      VB=V
      WB=W
C
C     ROTATE VB,WB IN SYSTEM 2 INTO V,W IN SYSTEM 1
C
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
C
C     INFLUENCE OF CORNER 1 ON MATCHING PANEL OF RIGHT WING
C
   11 YW=-(YI+YLC(I))
C
C     TRANSFORM YW,ZW IN SYSTEM 1 TO Y,Z, IN SYSTEM 2
C
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTA)
      IF(.NOT.FELTA)) GO TO 10
C
C     RETURN U TO SYSTEM 1
C
      U=-U
      VB=V
      WB=W
C
C     TRANSFORM VB,WB IN SYSTEM 2 TO V,W IN SYSTEM 1
C
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
      GO TO 10
C
C     CORNER 1 EQUATIONS FOR PANELS IN UPPER LEFT QUADRANT
C
   15 X=XLF(I)-XI
      YW=YI-YLC(I)
      ZW=ZI-ZLC(I)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 16
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   16 YW=-(YLC(I)+YI)
      Y=YW*CS+ZW*SN
```

Figure A-1(kk)

```
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTA1)
      IF(.NOT.FELTA1) GO TO 17
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
      GO TO 17
C
C     ********** CORNER 2 ********************************************
C     EQUATIONS FOR PANELS IN LOWER LEFT QUADRANT
C
   10 X=XRF(I)-XI
      YW=YI-YRC(I)
      ZW=ZI-ZRC(I)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTB)
      IF(.NOT. FELTB) GO TO 12
      U=-U
      VB=V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   12 YW=-(YI+YRC(I))
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTB1)
      IF(.NOT.FELTB1) GO TO 13
      U=-U
      VB=V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
   13 IF(.NOT.FELT) GO TO 40
      GO TO 20
C
C     CORNER 2 EQUATIONS FOR PANELS IN THE UPPER LEFT QUADRANT
C
   17 X=XRF(I)-XI
      YW=YI-YRC(I)
      ZW=ZI-ZRC(I)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 24
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   24 YW=-(YRC(I)+YI)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTB1)
      IF(.NOT.FELTA1) GO TO 25
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
   25 IF(.NOT.FELT) GO TO 40
C
C     ******************** CORNER 3 ***********************************
C
   20 IF(XI.GE.XLA(I)) GO TO 40
      IF((THTI(K).GT.90..AND.THTI(K).LE.180.0)) GO TO 19
C
C     CORNER 3 EQUATIONS FOR PANELS IN THE LOWER LEFT QUADRANT
C
      X=XLB(I)-XI
      YW=YI-YLC(I)
      ZW=ZI-ZLC(I)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 23
      U=-U
      VB=V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV-V
      TW=TW-W
   23 YW=-(YI+YLC(I))
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTA1)
      IF(.NOT.FELTA1) GO TO 30
      U=-U
      VB=V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV+V
      TW=TW-W
      GO TO 30
C
C     CORNER 3 EQUATIONS FOR PANELS IN UPPER LEFT QUADRANT
C
   19 X=XLB(I)-XI
      YW=YI-YLC(I)
      ZW=ZI-ZLC(I)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 26
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV+V
      TW=TW+W
   26 YW=-(YLC(I)+YI)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTA1)
      IF(.NOT.FELTA1) GO TO 21
```

Figure A-1(11)

```
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV-V
      TW=TW+W
      GO TO 21
C
C     ******************** CORNER 4 ************************************
C     CORNER 4 EQUATIONS FOR PANELS IN THE LOWER LEFT QUADRANT
C
   30 X=XRB(I)-XI
      YW=YI-YRC(I)
      ZW=ZI-ZRC(I)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 35
      U=-U
      VB=V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV+V
      TW=TW+W
   35 YW=-(YI+YRC(I))
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      CALL VELO1(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      U=-U
      VB=V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU+U
      TV=TV-V
      TW=TW+W
      GO TO 40
C
C     CORNER 4 EQUATIONS FOR PANELS IN UPPER LEFT QUADRANT
C
   21 X=XRB(I)-XI
      YW=YI-YRC(I)
      ZW=ZI-ZRC(I)
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 28
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV-V
      TW=TW-W
   28 YW=-(YI+YRC(I))
      Y=YW*CS+ZW*SN
      Z=YW*SN-ZW*CS
      Y=-Y
      CALL VELO1(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      U=-U
      VB=-V
      WB=W
      V=VB*CS+WB*SN
      W=VB*SN-WB*CS
      TU=TU-U
      TV=TV+V
      TW=TW-W
C
C     RETURN INFLUENCE COEFFICIENTS
C
   40 UP=TU
      VP=TV
      WP=TW
      RETURN
      END


      SUBROUTINE VELC( (T,TC,TX,N,XP,Y,ZP,U1,V1,W1,BODYL,SUMR,SUMRD)
C
C SUBROUTINE TO CALCULATE THE VELOCITIES FOR THE FIELD POINT X,Y,Z DUE
C TO THE BODY SINGULARITIES.
C
C     XP,Y,ZP - FIELD POINT COORDINATES WHERE VELOCITY
C               IS TO BE CALCULATED
C     XP,Y,ZP IN BODY SYSTEM
C          XP - POSITIVE FORWARD
C          Y  - POSITIVE RIGHT
C          ZP - POSITIVE DOWN
C     U1,V1,W1 - VELOCITIES ACCORDING TO THE FOLLOWING SIGN CONVENTION
C           U1 - POSITIVE FORWARD
C           V1 - POSITIVE TO RIGHT
C           W1 - POSITIVE DOWN
C
      DIMENSION T(100),TC(100),TX(101)
C
      COMMON/FLOW/ DUM1(5),BETA,BETASQ,DUM2
      COMMON/SRCE/XFIELD,RFIELD,BSQ,X2,      U,V,VT
C
C     TRANSFORM FIELD POINT COORDINATES TO VELCAL SYSTEM
C
      X=-XP
      Z=-ZP
      XFIELD=X
      RFIELD=SQRT(Y*Y+Z*Z)
      THETA=ATAN2(Y,Z)
C
C CALCULATION OF AXIAL, RADIAL AND TANGENTIAL PERTURBATION VELOCITIES
C
      US=0.
      VS=0.
      UD=0.
      VD=0.
      VTD=0.
      DO 110 J=1,N
      X1=XFIELD-TX(J)
      BR=BETA*RFIELD
      IF(X1.LE.BR) GO TO 115
      XL=X1/BR
      D23=SQRT(XL*XL-1.)
      ACOSH=ALOG(XL+D23)
      U=-ACOSH
      V=BETA*D23
      US=US+T(J)*U
      VS=VS+T(J)*V
      U=V
      XX=XL*D23
      V=-.5*BETASQ*(ACOSH+XX)
      VT=.5*BETASQ*(ACOSH-XX)
      UD=UD+U*TC(J)
      VD=VD+V*TC(J)
  110 VTD=VTD+VT*TC(J)
C
C     ADD VELOCITIES DUE TO SINGULARITIES STARTING AT BODY BASE
C
      X1=XFIELD-BODYL
      BR=BETA*RFIELD
      IF (X1.LE.BR) GO TO 115
      XL=X1/BR
```

Figure A-1(mm)

```
      D23=SQRT(XL*XL-1.0)
      ACOSH=ALOG(XL+D23)
      US=US+SUMK*ACOSH
      VS=VS-SUMK*BETA*D23
      UD=UD-BETA*D23*SUMKD
      VD=VD+0.5*BETASQ*(XL*D23+ACOSH)*SUMKD
      VTD=VTD+0.5*BETASQ*(XL*D23-ACOSH)*SUMKD
C
C TRANSFORMATION OF VELOCITIES INTO ORIGINAL COORDINATE SYSTEM...I.E....
C U1,V1,W1 RATHER THAN U,VR,VTHETA
C
  115 CONTINUE
      COSTH=COS(THETA)
      U1=-US-UD*COSTH
      VR=VD*COSTH+VS
      SINTH=SIN(THETA)
      VTD=VTD*SINTH
      V1=VR*SINTH+VTD*COSTH
      W1=VTD*SINTH-VR*COSTH
      RETURN
      END


      SUBROUTINE VELCMP(IREAD)
C
C  COMPUTE THE VELOCITY COMPONENTS (U,V,W) AT THE PANEL CONTROL
C  POINTS AND FORM THE AERODYNAMIC INFLUENCE COEFFICIENT MATRICES
C
C  TAPE USAGE IN VELCMP-
C    TAPE5  READ XMACH,ALPHA,PHIR
C
C    TAPE8  VELCMP,  REWIND
C           BODVEL,  WRITE  UB,VB,WB - COMPONENT VELOCITY MATRIICES
C           VELCMP,  REWIND
C
C    TAPE9  VELCMP,  REWIND
C           BODVEL,  WRITE  AN    - NORMAL VELOCITY MATRIX
C           VELCMP,  REWIND
C
C  BLANK COMMON USAGE FOR VELCMP
C
C  (1) AFTER READING EXTERNAL GEOMETRY FROM TAPE7
C        XPT,YPT,ZPT,THET,DELTA,XC,YC,ZC,...
C  (2) INCLUDING AERO MATRICES
C        XPT,YPT,ZPT,THET,DELTA,XC,YC,ZC,AN,UVWB,...
C
C
      COMMON /BODCOM/ AMACH,TANO,CX,XCOR(4),YCOR(4),ZCOR(4)
     1    ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,J
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPHES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XBLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NADIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *    ,IAR,IAN,IUB,ISB(7),IVR,IU,IV,IW,IVA,IWA,ICP,IPHI,IYR
     *    ,NAG,NAP,NAV,NAS,NASHX,NAFLD,IAO,IUO,ISKO,IZZ(4),NRING,IROW(50)
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1    ,SINAC,COSAC,SPHI,CPHI,SINA,SINH
      COMMON          A(12000)
C
      LOGICAL SUBSON,SUPERS
C
C  READ IN MACH NUMBER AND ANGLE OF ATTACK
C
      IF (IREAD.LE.0) READ (5,290) XMACH,ALPHAC,PHIR
      ALPHA=ALPHAC
      IF (XMACH.LT.0..OR.XMACH.EQ.EM) RETURN
      REWIND 8
      REWIND 9
      CALL IOWRIT (9,IROW,NRING)
      EM=XMACH
      AMACH=XMACH
      SUBSON = AMACH.LT.1.
      SUPERS = AMACH.GT.1.
      BETAO = SQRT(ABS(XMACH*XMACH-1.))
      BETAL = BETAO
      IAN = NG+1
      IUB = IAN+MAXKR**2
      NAV = IUB+3*MAXKR**2
      NATOT = MAXO(NAV,NATOT)
C
C  COMPUTE VELOCITY COMPONENTS INDUCED BY BODY PANELS
C              LOOP ON NUMBER OF SEGMENTS
C
      DO 110 IFU=1,KFUS
      IX = IXC(IFU)
      IY = IYC(IFU)
      IZ = IZC(IFU)
C
      CALL BODVEL (A(1),A(IYPT),A(IZPT),A(ITH),A(IDEL),A(IAN)
     *    ,A(IUB),A(IX),A(IY),A(IZ),KFORX(IFU),IFU,IXZSYM)
  110 CONTINUE
      REWIND 8
      REWIND 9
      RETURN
  290 FORMAT (10F7.0)
      END


      SUBROUTINE VELO1(FELT)
C
C     THIS SUBROUTINE CALCULATES THE INFLUENCE OF A
C     SEMI-INFINITE TRIANGLE UNDER CONSTANT U DIFFERENCE, U+/VINF
C
      LOGICAL PYPNL,FELT,INSIDE
C
      COMMON/CONSTS/PI,PI2,DTOR,RTOD,FOURPI
      COMMON /FLOW/ALFACR,GAMF,FMACH,RHO,VINF,BETA,BTSQ,FMCHSQ
      COMMON /VELARG/X,YS,ZS,U,V,W,EML,TLRNC,TIPY,PYPNL
C
C     FUNCTION STATEMENTS FOR COMPUTING F1,F4,F5,F7
C
      FN1(X)=ATAN2(Z*ROOTAL,EML*YZSQ-X*Y)
      FN4(Y)=(Y/YZSQ)*ROOTAL
      FN5(X)=ALOG((X+ROOTAL)/(BETA*SQRT(YZSQ)))
C
      FELT=.TRUE.
      IF(X.GE.TLRNC) GO TO 5
C
C     SET VELOCITIES EQUAL TO ZERO FOR WHATEVER REASON
C
    1 U=0.
      V=0.
      W=0.
      FELT=.FALSE.
      RETURN
C
C     CHECK IF INFLUENCING PANEL IS ON THE PYLON.
C     IF SO, Y=ZS AND Z=-YS
C
    5 IF(PYPNL) GO TO 10
      Y=YS
      Z=ZS
      GO TO 15
   10 Y=ZS
      Z=-YS
   15 YSQ=Y*Y
      ZSQ=Z*Z
      YZSQ=YSQ+ZSQ
      EMLSQ=EML*EML
      ARG1=X*X-BTSQ*YZSQ
      INSIDE=ARG1.GT.TLRNC
```

Figure A-1(nn)

```
C
C     CHECK FOR UNSWEPT LEADING EDGE
C
      IF(ABS(EML).LT.TLRNC) GO TO 40
      YYEDGE=Y-X/EML
C
C     CHECK WHETHER LEADING EDGE IS SUBSONIC,SONIC, OR SUPERSONIC.
C
      STEST=BTSQ-EMLSQ
      IF(ABS(STEST).LT.TLRNC) GO TO 20
      IF(STEST.GT.0.) GO TO 30
C
C************** SUBSONIC LEADING EDGE ***********************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF IT LIES OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C
      IF(.NOT.INSIDE) GO TO 1
C
C     INSIDE CONE FROM ORIGIN.
C
C     TEST FOR POSSIBLE SINGULARITIES IN FUNCTIONS.
C
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      IF(ABS(YYEDGE).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      ROOTA1=SQRT(ARG1)
      T1=X*EML-BTSQ*Y
      T2=BTSQ*((Y*EML-X)**2+ZSQ*(EMLSQ-BTSQ))
      F2=(EML/SQRT(EMLSQ-BTSQ))*ALOG((T1+SQRT(T1*T1-T2))/SQRT(T2))
      F1=FN1(X)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
      GO TO 100
C
C************** SONIC LEADING EDGE **************************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C     AT X-AXIS OR LEADING EDGE SET U,V,W TO ZERO
C
   20 IF(.NOT.INSIDE) GO TO 1
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      IF(ABS(YYEDGE).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      ROOTA1=SQRT(ARG1)
      F2=ROOTA1/(X-BETA*Y)
      F1=FN1(X)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
      GO TO 100
C
C
C************** SUPERSONIC LEADING EDGE *********************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO MAKE.
C
   30 IF(INSIDE) GO TO 35
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN.
C     DETERMINE IF IT IS INSIDE CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C
      IF(Y.LT.0.0.OR.YYEDGE.GE.0.0) GO TO 1
      YC=X*EML/BTSQ
      IF (Y.LT.YC) GO TO 1
      XLE=Y*EML
      XTRNSF=X-XLE
      ZCONE=XTRNSF/SQRT(STEST)
      IF (ABS(Z).GT.ZCONE) GO TO 1
C
C     INSIDE MACH CONE FROM LEADING EDGE
C
      F1=PI
      IF(Z.LT.0.0) F1=-PI
      F2=PI*EML/SQRT(BTSQ-EMLSQ)
      F4=0.
      F5=0.
      F7=0.
      GO TO 100
C
C     INSIDE MACH CONE FROM ORIGIN
C     AT X-AXIS OR LEADING SET U,V,W TO ZERO
C
   35 CONTINUE
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      IF(ABS(YYEDGE).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      ROOTA1=SQRT(ARG1)
      T3=SQRT(BTSQ-EMLSQ)
      F2=(EML/T3)*ATAN2(T3*ROOTA1,X*EML-BTSQ*Y)
      F1=FN1(X)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
      GO TO 100
C
C
C************** UNSWEPT LEADING EDGE ************************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO MAKE
C
   40 IF(INSIDE) GO TO 45
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN.
C     DETERMINE IF IT IS INSIDE CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C
      IF(Y.LT.0.0) GO TO 1
      RSTB=ABS(Z)*BETA
      IF(X.LE.RSTB) GO TO 1
C
C     INSIDE MACH CONE FROM LEADING EDGE
C
      F1=PI
      IF(Z.LT.0.0) F1=-PI
      F2=-BETA*PI
      F4=0.
      F5=0.
      F7=0.
      GO TO 48
C
C     INSIDE MACH CONE FROM ORIGIN
C
   45 CONTINUE
      ROOTA1=SQRT(ARG1)
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      F1=ATAN2(Z*ROOTA1,-X*Y)
      F2=-BETA*ATAN2(ROOTA1,-BETA*Y)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
C
C     VELOCITIES FOR UNSWEPT LE CASE
C
   48 U=F1
      V=F7
      W=F2-F4
      GO TO 105
C
C************** CALCULATE PERTURBATION VELOCITIES U,V,W *****************
C
  100 U=F1
      V=-EML*F1+F7
      W=EML*((1.0-BTSQ/EMLSQ)*F2-F5)-F4
C
C     TRANSFORM V AND W FOR PYLON PANELS
```

Figure A-1(oo)

```
C
  195 IF(.NOT. PYPNL) GO TO 110
      TEMP=V
      V=-W
      W=TEMP
  110 CONTINUE
      RETURN
      END


      SUBROUTINE VELOT1(FELT)
C
C     THIS SUBROUTINE CALCULATES THE INFLUENCE OF A SEMI-INFINITE
C     TRIANGLE ASSOCIATED WITH A THICKNESS SOURCE PANEL
C     THE SUBROUTINE IS USED ONLY TO CALCULATE INFLUENCE
C     FUNCTIONS AT A CONSTANT U-VELOCITY CONTROL POINT.
C
      LOGICAL PYPNL,INSIDE,FELT
      COMMON/CONSTS/PI,PI2,DTOR,HI00,FOURPI
      COMMON /FLOW/ALFACR,GAMF,FMACH,RHO,VINF,BETA,BTSQ,FMCHSQ
      COMMON /THVARG/X,YTH,ZTH,UTH,VTH,WTH,EML,PRT
      COMMON /VELARG/DUM(7),TLRNC,TIPY,PYPNL
C
C
      FELT=.TRUE.
      IF(X.GE.TLRNC) GO TO 3
C
C     SET VELOCITIES TO ZERO FOR WHATEVER REASON
C
    1 UTH=0.0
      VTH=0.0
      WTH=0.0
      FELT=.FALSE.
      RETURN
C     .
C     CHECK IF THE INFLUENCING PANEL IS ON THE PYLON
C     IF SO, Y=ZTH AND Z=-YTH
C
    3 IF(PYPNL) GO TO 4
      Y=YTH
      Z=ZTH
      GO TO 5
    4 Y=ZTH
      Z=-YTH
    5 CONTINUE
      IF(ABS(Z).LT.TLRNC) GO TO 1
      YSQ=Y*Y
      ZSQ=Z*Z
      EMLSQ=EML*EML
      ARG1=X*X-BTSQ*(YSQ+ZSQ)
      INSIDE=ARG1 .GT. TLRNC
C
C     CHECK FOR SPECIAL CASE OF UNSWEPT LEADING EDGE
C
      IF(ABS(EML).LT.TLRNC) GO TO 70
      YEDGE=X/EML
      ARGY=Y-YEDGE
C
C     CHECK WHETHER LEADING EDGE IS SUBSONIC, SONIC, OR SUPERSONIC
C
      STEST=BTSQ-EMLSQ
      IF(ABS(STEST).LT.TLRNC) GO TO 20
      IF(STEST.GT.0.0) GO TO 30
C
C********** SUBSONIC LEADING EDGE **********
C
C     DETERMINE IF POINT LIES INSIDE OR OUTSIDE MACH CONE FROM ORIGIN
C     IF IT LIES OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO
C
      IF(.NOT.INSIDE) GO TO 1
C
C     POINT LIES INSIDE MACH CONE FROM ORIGIN
C
      RAD=SQRT(ARG1)
      T1=X*EML-BTSQ*Y
      T2=BETA*SQRT((Y*EML-X)*(Y*EML-X)+ZSQ*(EMLSQ-BTSQ))
C
C     TEST FOR POSSIBLE SINGULARITY IN F1 OR F2
C
      IF(ABS(ARGY).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      F2=(EML/SQRT(EMLSQ-BTSQ))*ALOG((T1+SQRT(T1*T1-T2*T2))/T2)
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 13
      F1=ATAN2(Z*RAD,EML*(YSQ+ZSQ)-Y*X)
      FS=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)) )
      GO TO 100
C
C     CASE FOR Y AND Z BOTH SMALL
C
   13 F1=0.0
      VTH=0.0
      GO TO 101
C
C********** SONIC LEADING EDGE **********
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO
C
   20 IF(.NOT.INSIDE) GO TO 1
      RAD=SQRT(ARG1)
      F2=RAD/(X-BETA*Y)
C
C     F1 AND FS SAME AS IN SUBSONIC LEADING EDGE
C
      IF (ABS(Y) .LT. TLRNC .AND. ABS(Z) .LT. TLRNC) GO TO 21
      FS=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)))
      IF(ABS(ARGY).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 22
      F1=ATAN2(Z*RAD,EML*(YSQ+ZSQ)-Y*X)
      GO TO 100
C
C     Y AND Z BOTH SMALL
C
   21 F1=0.
      VTH = 0.
      GO TO 101
C
C     Z SMALL AND Y CLOSE TO LEADING EDGE
C
   22 F1=0.
      GO TO 100
C
C********** SUPERSONIC LEADING EDGE **********
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO BE MADE
C
   30 IF(INSIDE) GO TO 31
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN
C     DETERMINE IF IT IS INSIDE MACH CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO
C
      IF(Y .LT. 0.0 .OR. Y .GE. YEDGE) GO TO 1
      YC=X*EML/BTSQ
      IF (Y.LT.YC) GO TO 1
      XLE=Y*EML
      XTRNSF=X-XLE
      ZCONE=XTRNSF/SQRT(STEST)
      IF (ABS(Z).GT.ZCONE) GO TO 1
C
C     POINT IS INSIDE MACH CONE FROM LEADING EDGE
C
      F2= PI*EML/SQRT(BTSQ-EMLSQ)
      F1=PI
      IF(Z.LT.0.0) F1=-PI
      FS=0.0
      GO TO 100
```

Figure A-1(pp)

```
C
C     POINT IS INSIDE MACH CONE FROM ORIGIN
C
   31 T3=SQRT(BTSQ-EMLSQ)
      RAD=SQRT(ARG1)
      F2=(EML/T3)*ATAN2(RAD*T3,X*EML-BTSQ*Y)
C
C     USE F1 AND F5 AS IN SUBSONIC LEADING EDGE CASE
C
      IF (ABS(Y) .LT. TLRNC .AND. ABS(Z) .LT. TLRNC) GO TO 32
      F5=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)))
      IF(ABS(ARGY).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 33
      F1=ATAN2(Z*RAD,EML*(YSQ+ZSQ)-Y*X)
      GO TO 100
C
C     Y AND Z BOTH SMALL
C
   32 F1=0.
      VTH=0.
      GO TO 101
C
C     Z SMALL AND Y CLOSE TO LEADING EDGE
C
   33 F1=0.
      GO TO 100
C
C********** SPECIAL CASE FOR UNSWEPT LEADING EDGE **********
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO BE MADE
C
   70 IF(INSIDE) GO TO 72
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN
C     DETERMINE IF IT IS INSIDE MACH CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO
C
      IF(Y .LT. 0.) GO TO 1
      RSTB=ABS(Z)*BETA
      XTEST=X-RSTB
      IF((RSTB .LT. TLRNC .AND. X.LE.0.).OR.XTEST.LE.TLRNC)GO TO 1
C
C     POINT BETWEEN MACH CONE FROM ORIGIN AND LEADING EDGE
C
      UTH=-PI/BETA
      F1=PI
      IF(Z.LT.0.0) F1=-PI
      VTH=0.0
      WTH=F1
      GO TO 102
C
C     POINT IS INSIDE MACH CONE FROM ORIGIN
C
   72 RAD=SQRT(ARG1)
      UTH=-ATAN2(RAD,-BETA*Y)/BETA
      F2=0.0
      IF(ABS(Z) .LT. TLRNC .AND. ABS(Y) .LT. TLRNC)GO TO 73
      F1=ATAN2(Z*RAD,-Y*X)
      F5=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)))
      VTH=-F5
      WTH=F1
      GO TO 102
   73 VTH=0
      WTH=0
      GO TO 102
C
C     COMPUTE PERTURBATION VELOCITIES
C
  100 VTH=F2-F5
  101 UTH=-F2/EML
      WTH=F1
  102 IF (.NOT. PYPNL) GO TO 103
      TEMP=VTH
      VTH=-WTH
      WTH=TEMP
  103 CONTINUE
      RETURN
      END



      SUBROUTINE VELPP1(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES THE INFLUENCE COEFFICIENTS DUE TO A
C     PYLON CONSTANT U VELOCITY PANEL AT THE CONTROL POINT OF
C     ANOTHER CONSTANT U VELOCITY PANEL
C     SUPERPOSITION OF 4 CORNER SOLUTIONS IS USED
C     INFLUENCE COEFFICIENTS UP,VP,WP ARE RETURNED IN THE WING
C     REFERENCE FRAME
C
      COMMON/BPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     1      SANGW
      COMMON/ICVEL/UP,VP,WP,I1,IF,          DELTP(200)
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON/PYGEOM/ZZZ(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
     1      SLLE,PSLPDF,CENTER,ZPL,LVSPP
      COMMON/WPGEOM/XWF(200),XRR(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2      SNPHI(200),CSPHI(200)
      COMMON /VELARG/X,Y,Z,U,V,W,EM,TLRNC,TIPY,PYPNL
C
      LOGICAL        PYPNL,FELTA,FELTA1,FELTB,FELTB1,FELT,CENTER
C
      PYPNL=.TRUE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      ZI=ZZ
      YDIR=YY-YPL
      YIMG=-(YY+YPL)
C
C     I IS INDEX OF INFLUENCING PANEL
C
      I=I1
      FELT=.FALSE.
      TU=0.0
      TV=0.0
      TW=0.0
      EM1=SWPPLE(I)
      EM2=SWPPTE(I)
C
C*******************  CORNER 1  ***********************************
C
      IF (EM1 .LT. 0.0) GO TO 15
C
C     CORNER 1 FOR EM1 .GE. 0 --- PANEL LEADING EDGE SWEPT BACK
C
      EM=EM1
      IF(XI.GE.XRF(I)) GO TO 40
      X=-XI+XLF(I)
      Y=YDIR
      Z=ZI-ZLC(I)
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 50
      TU=TU-U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
   50 IF(CENTER) GO TO 10
      Y=YIMG
      CALL VELO1(FELTA1)
      IF(.NOT.FELTA1) GO TO 10
      TU=TU-U
      TV=TV+V
      TW=TW+W
```

Figure A-1(qq)

```
      FELT=.TRUE.
      GO TO 10
C
C     CORNER 1 FOR EM1 .LT. 0 --- PANEL LEADING EDGE SWEPT FORWARD
C
   15 CONTINUE
      IF(XI.GE.XLF(I)) GO TO 40
      EM=-EM1
      X=-XI+XLF(I)
      Y=YDIR
      Z=-ZI+ZLC(I)
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 52
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   52 IF(CENTER) GO TO 17
      Y=YIMG
      CALL VELO1(FELTAI)
      IF(.NOT.FELTAI) GO TO 17
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
      GO TO 17
C
C******************** CORNER 2 ***************************************
C
C     CORNER 2 FOR EM1 .GE. 0
C
   10 CONTINUE
      X=-XI+XRF(I)
      Y=YDIR
      Z=ZI-ZRC(I)
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 12
      TU=TU+U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
   12 IF(CENTER) GO TO 13
      Y=YIMG
      CALL VELO1(FELTBI)
      IF(.NOT.FELTBI)GO TO 13
      TU=TU+U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   13 IF(.NOT.FELT) GO TO 40
      GO TO 20
C
C     CORNER 2 FOR EM1 .LT. 0
C
   17 CONTINUE
      X=-XI+XRF(I)
      Y=YDIR
      Z=-ZI+ZRC(I)
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 18
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   18 IF(CENTER) GO TO 16
      Y=YIMG
      CALL VELO1(FELTBI)
      IF(.NOT.FELTBI) GO TO 16
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
   16 IF(.NOT.FELT) GO TO 40
C
C******************** CORNER 3 ***************************************
C
   20 IF (EM2 .LT. 0.0) GO TO 19
C
C     CORNER 3 FOR EM2 .GE. 0 --- PANEL TRAILING EDGE SWEPT BACK
C
      IF(XI.GE.XRB(I)) GO TO 40
      EM=EM2
      X=-XI+XLB(I)
      Y=YDIR
      Z=ZI-ZLC(I)
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 51
      TU=TU+U
      TV=TV+V
      TW=TW-W
   51 IF(CENTER) GO TO 30
      Y=YIMG
      CALL VELO1(FELTAI)
      IF(.NOT.FELTAI) GO TO 30
      TU=TU+U
      TV=TV-V
      TW=TW-W
      GO TO 30
C
C     CORNER 3 FOR EM2 .LT. 0 --- PANEL TRAILING EDGE SWEPT FORWARD
C
   19 EM=-EM2
      IF(XI.GE.XLB(I)) GO TO 40
      X=-XI+XLB(I)
      Y=YDIR
      Z=-ZI+ZLC(I)
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 53
      TU=TU-U
      TV=TV-V
      TW=TW-W
   53 IF(CENTER) GO TO 21
      Y=YIMG
      CALL VELO1(FELTAI)
      IF(.NOT.FELTAI) GO TO 21
      TU=TU-U
      TV=TV+V
      TW=TW-W
      GO TO 21
C
C******************** CORNER 4 ***************************************
C
C     CORNER 4 FOR EM2 .GE. 0
C
   30 CONTINUE
      X=-XI+XRB(I)
      Y=YDIR
      Z=ZI-ZRC(I)
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 35
      TU=TU-U
      TV=TV-V
      TW=TW+W
   35 IF(CENTER) GO TO 40
      Y=YIMG
      CALL VELO1(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      TU=TU-U
      TV=TV+V
      TW=TW+W
      GO TO 40
C
C     CORNER 4 FOR EM2 .LT. 0
C
   21 CONTINUE
      X=-XI+XRB(I)
      Y=YDIR
      Z=-ZI+ZRC(I)
```

Figure A-1(rr)

```
      CALL VELOI(FELTB)
      IF(.NOT.FELTB) GO TO 25
      TU=TU+U
      TV=TV+V
      TW=TW+W
   25 IF(CENTER) GO TO 40
      Y=YIMG
      CALL VELOI(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      TU=TU+U
      TV=TV-V
      TW=TW+W
C
C     SIGN CHANGES NECESSARY TO ACCOMODATE USE OF SUBROUTINE
C     VELO WITH DIFFERENT COORDINATE SYSTEM
C
   40 CONTINUE
      TU=-TU
      TW=-TW
C
C     RETURN INFLUENCE COEFFICIENTS
C
      UP=TU
      VP=TV
      WP=TW
      RETURN
      END


      SUBROUTINE VELPTI(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED BY
C     THE PYLON THICKNESS PANELS.
C     SUPERPOSITION OF 4 CORNER SOLUTIONS IS USED
C     VELPTH RETURNS VELOCITIES UP,VP,WP IN WING REFERENCE FRAME.
C
      COMMON /ICVEL/UP,VP,WP,II,IF,        DELTP(200)
      COMMON /PYGEOM/ZZZ(20),XPLE,YPL,CHP,HP,PSIPLE(20),PSIPTE(20),IP,
     1      SLLE,PSLPOF,CENTER,ZPL,LVSPP
      COMMON/THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     1   XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2   THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3     YS(20),PSWSLF(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4     PSPSTE(20),SNPHS(400),CSPHS(400)
      COMMON /THVARG/XTH,YTH,ZTH,UTH,VTH,WTH,EML,PRT
      COMMON /VELARG/DUMY(9),PYPNL
C
      EQUIVALENCE (X,XTH),(Y,YTH),(Z,ZTH),(U,UTH),(V,VTH),(W,WTH)
C
      LOGICAL FELTA,FELTAI,FELTB,FFLTBI,FELT,       PYPNL,CENTER
C
      PYPNL=.TRUE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      ZI=ZZ
      YDIR=YY-YPL
      YIMG=-(YY+YPL)
C
C     I IS INDEX OF INFLUENCING PANEL
C
      DO 100 I=II,IF
      FELT=.FALSE.
      TU=0.0
      TV=0.0
      TW=0.0
      EM1=SLLET(I)
      EM2=SLTET(I)
C
C******************  CORNER 1  ******************************************
C
      IF (EM1 .LT. 0.0) GO TO 15
C
C     CORNER 1 FOR EM1.GE. 0 --- PANEL LEADING EDGE SWEPT BACK
C
      IF(XI.GE.XRFT(I)) GO TO 100
      EML=EM1
      X=-XI+XLFT(I)
      Y=YDIR
      Z=ZI-ZLCT(I)
      CALL VELOTI(FELTA)
      IF(.NOT.FELTA) GO TO 11
      TU=TU-U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
   11 IF(CENTER) GO TO 10
      Y=YIMG
      CALL VELOTI(FELTAI)
      IF(.NOT.FELTAI) GO TO 10
      TU=TU-U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
      GO TO 10
C
C     CORNER 1 FOR EM1.LT. 0 --- PANEL LEADING EDGE SWEPT FORWARD
C
   15 CONTINUE
      IF(XI.GE.XLFT(I)) GO TO 100
      EML=-EM1
      X=-XI+XLFT(I)
      Y=YDIR
      Z=-ZI+ZLCT(I)
      CALL VELOTI(FELTA)
      IF(.NOT.FELTA) GO TO 16
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   16 IF(CENTER) GO TO 17
      Y=YIMG
      CALL VELOTI(FELTAI)
      IF(.NOT.FELTAI) GO TO 17
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
      GO TO 17
C
C******************  CORNER 2  ******************************************
C
C     CORNER 2 FOR EM1 .GE. 0
C
   10 CONTINUE
      X=-XI+XRFT(I)
      Y=YDIR
      Z=ZI-ZRCT(I)
      CALL VELOTI(FELTB)
      IF(.NOT.FELTB) GO TO 12
      TU=TU+U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
   12 IF(CENTER) GO TO 13
      Y=YIMG
      CALL VELOTI(FELTBI)
      IF(.NOT.FELTBI) GO TO 13
      TU=TU+U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   13 IF(.NOT.FELT) GO TO 100
      GO TO 20
C
```

Figure A-1(ss)

```
C     CORNER 2 FOR EM1 .LT. 0
C
   17 CONTINUE
      X=-XI+XRFT(I)
      Y=YDIR
      Z=-ZI+ZRCT(I)
      CALL VELOT1(FELTB)
      IF(.NOT.FELTB) GO TO 24
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   24 IF(CENTER) GO TO 25
      Y=YIMG
      CALL VELOT1(FELTBI)
      IF(.NOT.FELTBI) GO TO 25
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
   25 IF(.NOT.FELT) GO TO 100
C
C******************** CORNER 3 ****************************************
C
   20 IF (EM2 .LT. 0.0) GO TO 19
C
C     CORNER 3 FOR EM2 .GE.0 --- PANEL TRAILING EDGE SWEPT BACK
C
      IF(XI.GE.XRRT(I)) GO TO 40
      EML=EM2
      X=-XI+XLBT(I)
      Y=YDIR
      Z=ZI-ZLCT(I)
      CALL VELOT1(FELTA)
      IF(.NOT.FELTA) GO TO 23
      TU=TU+U
      TV=TV+V
      TW=TW-W
   23 IF(CENTER) GO TO 30
      Y=YIMG
      CALL VELOT1(FELTAI)
      IF(.NOT.FELTAI) GO TO 30
      TU=TU+U
      TV=TV-V
      TW=TW-W
      GO TO 30
C
C     CORNER 3 FOR EM2 .LT.0 --- PANEL TRAILING EDGE SWEPT FORWARD
C
   19 EML=-EM2
      IF(XI.GE.XLBT(I)) GO TO 40
      X=-XI+XLBT(I)
      Y=YDIR
      Z=-ZI+ZLCT(I)
      CALL VELOT1(FELTA)
      IF(.NOT.FELTA) GO TO 26
      TU=TU-U
      TV=TV-V
      TW=TW-W
   26 IF(CENTER) GO TO 21
      Y=YIMG
      CALL VELOT1(FELTAI)
      IF(.NOT.FELTAI) GO TO 21
      TU=TU-U
      TV=TV+V
      TW=TW-W
      GO TO 21
C
C******************** CORNER 4 ****************************************
C
C     CORNER 4 FOR EM2 .GE. 0
C
   30 CONTINUE
      X=-XI+XRBT(I)
      Y=YDIR
      Z=ZI-ZRCT(I)
      CALL VELOT1(FELTB)
      IF(.NOT.FELTB) GO TO 35
      TU=TU-U
      TV=TV-V
      TW=TW+W
   35 IF(CENTER) GO TO 40
      Y=YIMG
      CALL VELOT1(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      TU=TU-U
      TV=TV+V
      TW=TW+W
      GO TO 40
C
C     CORNER 4 FOR EM2 .LT. 0
C
   21 CONTINUE
      X=-XI+XRBT(I)
      Y=YDIR
      Z=-ZI+ZRCT(I)
      CALL VELOT1(FELTB)
      IF(.NOT.FELTB) GO TO 28
      TU=TU+U
      TV=TV+V
      TW=TW+W
   28 IF(CENTER) GO TO 40
      Y=YIMG
      CALL VELOT1(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      TU=TU+U
      TV=TV-V
      TW=TW+W
C
C     SIGN CHANGES NECESSARY TO ACCOMMODATE USE OF SUBROUTINE
C     VELOTH WITH DIFFERENT COORDINATE SYSTEM
C
   40 CONTINUE
      TU=-TU
      TW=-TW
C
C     CALCULATION OF PERTURBATION VELOCITIES
C
      UP=UP+TU*DZDX(I)
      VP=VP+TV*DZDX(I)
      WP=WP+TW*DZDX(I)
  100 CONTINUE
      RETURN
      END


      SUBROUTINE VELWP1(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES THE INFLUENCE COEFFICIENTS DUE TO A
C     WING CONSTANT U VELOCITY PANEL AT THE CONTROL POINT OF ANOTHER
C     CONSTANT U VELOCITY PANEL
C     SUPERPOSITION OF 4 CORNER SOLUTIONS IS USED
C     INFLUENCE COEFFICIENTS UP,VP,WP ARE RETURNED IN THE WING
C     REFERENCE FRAME
C
      COMMON/BPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),SNGW
      COMMON/ICVEL/UP,VP,WP,II,IF,            DELTP(200)
      COMMON /WGEOM/XRWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),PSIWTE(20),
     1              DUM(20),PHID(20),ZDIHED,WICR
      COMMON/WPGEOM/XRF(200),XRR(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2   SNPHI(200),CSPHI(200)
      COMMON /VELARG/X,YV,ZV,U,VV,WV,EM,TLANC,TIPY,PYPNL
C
      LOGICAL PYPNL,FELT,FELTA,FELTAI,FELTB,FELTBI,ZDIHED
C
```

Figure A-1(tt)

```
C     STATEMENT FUNCTIONS FOR COORDINATE AND VELOCITY ROTATIONS
C
      ROTA(A,B)=A*CSPH-B*SNPH
      ROTB(A,B)=A*SNPH+B*CSPH
      ROTAC(A,B)=A*CSPH+B*SNPH
      ROTBC(A,B)=-A*SNPH+B*CSPH
C
      PYPNL=.FALSE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      YI=YY
      ZI=ZZ
C
C     I IS INDEX OF INFLUENCING PANEL
C
      I=II
      FELT=.FALSE.
      TU=0.0
      TV=0.0
      TW=0.0
      EM1=SWPPLE(I)
      EM2=SWPPTE(I)
      CSPH=CSPHI(I)
      SNPH=SNPHI(I)
C
C*******************  CORNER 1  ********************************* *****
C
      IF (EM1 .LT. 0.0) GO TO 15
C
C     CORNER 1 FOR EM1 .GE. 0 ---PANEL LEADING EDGE SWEPT BACK
C
      IF(XI.GE.XRF(I)) GO TO 40
      EM=EM1
      ICR=1
      X=-XI+XLF(I)
      Y=-YI+YLC(I)
      Z=-ZI+ZLC(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 60
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
   60 CONTINUE
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 50
      V=VV
      W=WV
      IF(ZDIMED) GO TO 62
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   62 CONTINUE
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
   50 Y=YI+YLC(I)
      YV=Y
      IF(ZDIMED) GO TO 64
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
   64 CONTINUE
      CALL VELO1(FELTA1)
      IF(.NOT.FELTA1) GO TO 10
      V=VV
      W=WV
      IF(ZDIMED) GO TO 66
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   66 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
      GO TO 10
C
C     CORNER 1 FOR EM1 .LT. 0 ---PANEL LEADING EDGE SWEPT FORWARD
C
   15 CONTINUE
      IF(XI.GE.XLF(I)) GO TO 40
      EM=-EM1
      ICR=1
      X=-XI+XLF(I)
      Y=YI-YLC(I)
      Z=-ZI+ZLC(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 70
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
   70 CONTINUE
      CALL VELO1(FELTA)
      IF(.NOT.FELTA) GO TO 52
      V=VV
      W=WV
      IF(ZDIMED) GO TO 72
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   72 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   52 Y=-(YI+YLC(I))
      YV=Y
      IF(ZDIMED) GO TO 74
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
   74 CONTINUE
      CALL VELO1(FELTA1)
      IF(.NOT.FELTA1) GO TO 17
      V=VV
      W=WV
      IF(ZDIMED) GO TO 76
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   76 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
      GO TO 17
C
C*******************  CORNER 2  ****************************************
C
C     CORNER 2 FOR EM1 .GE. 0
C
   10 CONTINUE
      ICR=2
      X=-XI+XRF(I)
      Y=-YI+YRC(I)
      Z=-ZI+ZRC(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 80
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
   80 CONTINUE
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 12
      V=VV
      W=WV
      IF(ZDIMED) GO TO 82
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   82 CONTINUE
      TU=TU+U
```

Figure A-1(uu)

```
      TV=TV-V                                                           VELW1690
      TW=TW-W                                                           VELW1700
      FELT=.TRUE.                                                       VELW1710
   12 Y=YI+YRC(I)                                                       VELW1720
      YV=Y                                                              VELW1730
      IF(ZDIMED) GO TO 84                                               VELW1740
      YV=ROTAC(Y,Z)                                                     VELW1750
      ZV=ROTBC(Y,Z)                                                     VELW1760
   84 CONTINUE                                                          VELW1770
      CALL VELO1(FELTB1)                                                VELW1780
      IF(.NOT.FELTB1)GO TO 13                                           VELW1790
      V=VV                                                              VELW1800
      W=WV                                                              VELW1810
      IF(ZDIMED) GO TO 86                                               VELW1820
      V=ROTA(VV,WV)                                                     VELW1830
      W=ROTB(VV,WV)                                                     VELW1840
   86 CONTINUE                                                          VELW1850
      TU=TU+U                                                           VELW1860
      TV=TV+V                                                           VELW1870
      TW=TW+W                                                           VELW1880
      FELT=.TRUE.                                                       VELW1890
   13 IF(.NOT.FELT) GO TO 40                                            VELW1900
      GO TO 20                                                          VELW1910
C                                                                       VELW1920
C     CORNER 2 FOR EM1 .LT. 0                                           VELW1930
C                                                                       VELW1940
   17 CONTINUE                                                          VELW1950
      ICR=2                                                             VELW1960
      X=-XI+XRF(I)                                                      VELW1970
      Y=YI-YRC(I)                                                       VELW1980
      Z=-ZI+ZRC(I)                                                      VELW1990
      ZV=Z                                                              VELW2000
      YV=Y                                                              VELW2010
      IF(ZDIMED) GO TO 90                                               VELW2020
      YV=ROTA(Y,Z)                                                      VELW2030
      ZV=ROTB(Y,Z)                                                      VELW2040
   90 CONTINUE                                                          VELW2050
      CALL VELO1(FELTB)                                                 VELW2060
      IF(.NOT.FELTB) GO TO 18                                           VELW2070
      V=VV                                                              VELW2080
      W=WV                                                              VELW2090
      IF(ZDIMED) GO TO 92                                               VELW2100
      V=ROTAC(VV,WV)                                                    VELW2110
      W=ROTBC(VV,WV)                                                    VELW2120
   92 CONTINUE                                                          VELW2130
      TU=TU-U                                                           VELW2140
      TV=TV-V                                                           VELW2150
      TW=TW-W                                                           VELW2160
      FELT=.TRUE.                                                       VELW2170
   18 Y=-(YI+YRC(I))                                                    VELW2180
      YV=Y                                                              VELW2190
      IF(ZDIMED) GO TO 94                                               VELW2200
      YV=ROTA(Y,Z)                                                      VELW2210
      ZV=ROTB(Y,Z)                                                      VELW2220
   94 CONTINUE                                                          VELW2230
      CALL VELO1(FELTB1)                                                VELW2240
      IF(.NOT.FELTB1) GO TO 16                                          VELW2250
      V=VV                                                              VELW2260
      W=WV                                                              VELW2270
      IF(ZDIMED) GO TO 96                                               VELW2280
      V=ROTAC(VV,WV)                                                    VELW2290
      W=ROTBC(VV,WV)                                                    VELW2300
   96 CONTINUE                                                          VELW2310
      TU=TU-U                                                           VELW2320
      TV=TV+V                                                           VELW2330
      TW=TW-W                                                           VELW2340
      FELT=.TRUE.                                                       VELW2350
   16 IF(.NOT.FELT) GO TO 40                                            VELW2360
C                                                                       VELW2370
C********************  CORNER 3  ***************************************VELW2380
C                                                                       VELW2390
   20 IF (EM2 .LT. 0.0) GO TO 19                                        VELW2400
C                                                                       VELW2410
C     CORNER 3 FOR EM2 .GE. 0 --- PANEL TRAILING EDGE SWEPT BACK        VELW2420
C                                                                       VELW2430
      IF(XI.GE.XRB(I)) GO TO 40                                         VELW2440
      ICR=3                                                             VELW2450
      EM=EM2                                                            VELW2460
      X=-XI+XLB(I)                                                      VELW2470
      Y=-YI+YLC(I)                                                      VELW2480
      Z=-ZI+ZLC(I)                                                      VELW2490
      ZV=Z                                                              VELW2500
      YV=Y                                                              VELW2510
      IF(ZDIMED) GO TO 102                                              VELW2520
      YV=ROTAC(Y,Z)                                                     VELW2530
      ZV=ROTBC(Y,Z)                                                     VELW2540
  102 CONTINUE                                                          VELW2550
      CALL VELO1(FELTA)                                                 VELW2560
      IF(.NOT.FELTA) GO TO 51                                           VELW2570
      V=VV                                                              VELW2580
      W=WV                                                              VELW2590
      IF(ZDIMED) GO TO 104                                              VELW2600
      V=ROTA(VV,WV)                                                     VELW2610
      W=ROTB(VV,WV)                                                     VELW2620
  104 CONTINUE                                                          VELW2630
      TU=TU+U                                                           VELW2640
      TV=TV-V                                                           VELW2650
      TW=TW+W                                                           VELW2660
   51 Y=YI+YLC(I)                                                       VELW2670
      YV=Y                                                              VELW2680
      IF(ZDIMED) GO TO 106                                              VELW2690
      YV=ROTAC(Y,Z)                                                     VELW2700
      ZV=ROTBC(Y,Z)                                                     VELW2710
  106 CONTINUE                                                          VELW2720
      CALL VELO1(FELTA1)                                                VELW2730
      IF(.NOT.FELTA1) GO TO 30                                          VELW2740
      V=VV                                                              VELW2750
      W=WV                                                              VELW2760
      IF(ZDIMED) GO TO 108                                              VELW2770
      V=ROTA(VV,WV)                                                     VELW2780
      W=ROTB(VV,WV)                                                     VELW2790
  108 CONTINUE                                                          VELW2800
      TU=TU+U                                                           VELW2810
      TV=TV+V                                                           VELW2820
      TW=TW+W                                                           VELW2830
      GO TO 30                                                          VELW2840
C                                                                       VELW2850
C     CORNER 3 FOR EM2 .LT. 0 --- PANEL TRAILING EDGE SWEPT FORWARD     VELW2860
C                                                                       VELW2870
   19 CONTINUE                                                          VELW2880
      ICR=3                                                             VELW2890
      IF(XI.GE.XLB(I)) GO TO 40                                         VELW2900
      EM=-EM2                                                           VELW2910
      X=-XI+XLB(I)                                                      VELW2920
      Y=YI-YLC(I)                                                       VELW2930
      Z=-ZI+ZLC(I)                                                      VELW2940
      ZV=Z                                                              VELW2950
      YV=Y                                                              VELW2960
      IF(ZDIMED) GO TO 110                                              VELW2970
      YV=ROTA(Y,Z)                                                      VELW2980
      ZV=ROTB(Y,Z)                                                      VELW2990
  110 CONTINUE                                                          VELW3000
      CALL VELO1(FELTA)                                                 VELW3010
      IF(.NOT.FELTA) GO TO 53                                           VELW3020
      V=VV                                                              VELW3030
      W=WV                                                              VELW3040
      IF(ZDIMED) GO TO 112                                              VELW3050
      V=ROTAC(VV,WV)                                                    VELW3060
      W=ROTBC(VV,WV)                                                    VELW3070
  112 CONTINUE                                                          VELW3080
      TU=TU-U                                                           VELW3090
      TV=TV-V                                                           VELW3100
      TW=TW-W                                                           VELW3110
   53 Y=-(YI+YLC(I))                                                    VELW3120
      YV=Y                                                              VELW3130
      IF(ZDIMED) GO TO 114                                              VELW3140
      YV=ROTA(Y,Z)                                                      VELW3150
      ZV=ROTB(Y,Z)                                                      VELW3160
```

Figure A-1(vv)

```
  114 CONTINUE
      CALL VELO1(FELTA1)
      IF(.NOT.FELTA1)'GO TO 21
      V=VV
      W=WV
      IF(ZDIMED) GO TO 116
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
  116 CONTINUE
      TU=TU-U
      TV=TV+V
      TW=TW-W
      GO TO 21
C
C******************* CORNER 4 ****************************************
C
C     CORNER 4 FOR EM2 .GE. 0
C
   30 CONTINUE
      ICR=4
      X=-XI+XRB(I)
      Y=-YI+YRC(I)
      Z=-ZI+ZRC(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 120
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
  120 CONTINUE
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 35
      V=VV
      W=WV
      IF(ZDIMED) GO TO 122
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
  122 CONTINUE
      TU=TU-U
      TV=TV+V
      TW=TW-W
   35 Y=YI+YRC(I)
      YV=Y
      IF(ZDIMED) GO TO 124
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
  124 CONTINUE
      CALL VELO1(FELTB1)
      IF(.NOT.FELTB1) GO TO 40
      V=VV
      W=WV
      IF(ZDIMED) GO TO 126
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
  126 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      GO TO 40
C
C     CORNER 4 FOR EM2 .LT. 0
C
   21 CONTINUE
      ICR=4
      X=-XI+XRB(I)
      Y=YI-YRC(I)
      Z=-ZI+ZRC(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 130
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
  130 CONTINUE
      CALL VELO1(FELTB)
      IF(.NOT.FELTB) GO TO 25
      V=VV
      W=WV
      IF(ZDIMED) GO TO 132
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
  132 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
   25 Y=-(YI+YRC(I))
      YV=Y
      IF(ZDIMED) GO TO 134
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
  134 CONTINUE
      CALL VELO1(FELTB1)
      IF(.NOT.FELTB1) GO TO 40
      V=VV
      W=WV
      IF(ZDIMED) GO TO 136
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
  136 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
C
C
C
C     RETURN INFLUENCE COEFFICIENTS
C     SIGN CHANGES NECESSARY TO ACCOMODATE USE OF SUBROUTINE
C     VELO WITH DIFFERENT COORDINATE SYSTEM
C
   40 CONTINUE
      UP=-TU
      VP=TV
      WP=-TW
      RETURN
      END


      SUBROUTINE VELWT1(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED BY
C     THE WING THICKNESS PANELS.
C     SUPERPOSITION OF 4 CORNER SOLUTIONS IS USED
C     VELWTH RETURNS VELOCITIES UP,VP,WP IN WING REFERENCE FRAME
C
      COMMON /ICVEL/UP,VP,WP,I1,IF,        DELTP(200)
      COMMON/THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     1  XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2  THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3  YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4  PSPSTE(20),SNPHS(400),CSPHS(400)
      COMMON/THVARG/X,YV,ZV,U,VV,WV,EML,PRT
      COMMON /VELARG/DUMY(9),PYPNL
      COMMON/WGEOM/XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),PSIWTE(20),
     1  YYY(20),PHID(20),ZDIMED,WICR
C
C
      LOGICAL FELT,FELTA,FELTA1,FELTB,FELTB1,PYPNL,ZDIMED
C
C     STATEMENT FUNCTIONS FOR COORDINATE AND VELOCITY ROTATIONS
C
      ROTA(A,B)=A*CSPH-B*SNPH
      ROTB(A,B)=A*SNPH+B*CSPH
      ROTAC(A,B)=A*CSPH+B*SNPH
      ROTBC(A,B)=-A*SNPH+B*CSPH
C
      PYPNL=.FALSE.
      UP=0.0
      VP=0.0
```

Figure A-1(ww)

```
      WP=0.0
      XI=XX
      YI=YY
      ZI=ZZ
C
C     I IS INDEX OF INFLUENCING PANEL
C
      DO 100 I=II,IF
      FELT=.FALSE.
      TU=0.0
      TV=0.0
      TW=0.0
      EM1=SLLET(I)
      EM2=SLTET(I)
      CSPH=CSPHS(I)
      SNPH=SNPHS(I)
C
C******************** CORNER 1 *************************************
C
      IF (EM1 .LT. 0.0) GO TO 15
C
C     CORNER 1 FOR EM1 .GE. 0 ---PANEL LEADING EDGE SWEPT BACK
C
      IF(XI.GE.XRFT(I))GO TO 100
      ICR=1
      EML=EM1
      X=-XI+XLFT(I)
      Y=-YI+YLCT(I)
      Z=-ZI+ZLCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 60
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
   60 CONTINUE
      CALL VELOT1(FELTA)
      IF(.NOT.FELTA) GO TO 11
      V=VV
      W=WV
      IF(ZDIMED) GO TO 62
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   62 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   11 Y=YI-YLCT(I)
      YV=Y
      IF(ZDIMED) GO TO 64
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
   64 CONTINUE
      CALL VELOT1(FELTA1)
      IF(.NOT.FELTA1) GO TO 10
      V=VV
      W=WV
      IF(ZDIMED) GO TO 66
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   66 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
      GO TO 10
C
C     CORNER 1 FOR EM1 .LT. 0 ---PANEL LEADING EDGE SWEPT FORWARD
C
   15 CONTINUE
      IF(XI.GE.XLFT(I)) GO TO 100
      ICR=1
      EML=-EM1
      X=-XI+XLFT(I)
      Y=YI-YLCT(I)
      Z=-ZI+ZLCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 70
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
   70 CONTINUE
      CALL VELOT1(FELTA)
      IF(.NOT.FELTA) GO TO 16
      V=VV
      W=WV
      IF(ZDIMED) GO TO 72
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   72 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   16 Y=-(YI+YLCT(I))
      YV=Y
      IF(ZDIMED) GO TO 74
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
   74 CONTINUE
      CALL VELOT1(FELTA1)
      IF(.NOT.FELTA1) GO TO 17
      V=VV
      W=WV
      IF(ZDIMED) GO TO 76
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   76 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
      GO TO 17
C
C******************** CORNER 2 *************************************
C
C     CORNER 2 FOR EM1 .GE. 0
C
   10 CONTINUE
      ICR=2
      X=-XI+XRFT(I)
      Y=-YI+YRCT(I)
      Z=-ZI+ZRCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 80
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
   80 CONTINUE
      CALL VELOT1(FELTB)
      IF(.NOT.FELTB) GO TO 12
      V=VV
      W=WV
      IF(ZDIMED) GO TO 82
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   82 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
   12 Y=YI+YRCT(I)
      YV=Y
      IF(ZDIMED) GO TO 84
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
   84 CONTINUE
      CALL VELOT1(FELTB1)
```

Figure A-1(xx)

```
      IF(.NOT.FELTBI) GO TO 13
      V=VV
      W=WV
      IF(ZDIMED) GO TO 86
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   86 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   13 IF(.NOT.FELT) GO TO 100
      GO TO 20
C
C     CORNER 2 FOR EM1 .LT. 0
C
   17 CONTINUE
      ICR=2
      X=-XI+XRFT(I)
      Y=YI-YRCT(I)
      Z=-ZI+ZRCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 90
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
   90 CONTINUE
      CALL VELOT1(FELTB)
      IF(.NOT.FELTB) GO TO 24
      V=VV
      W=WV
      IF(ZDIMED) GO TO 92
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   92 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   24 Y=-(YI+YRCT(I))
      YV=Y
      IF(ZDIMED) GO TO 94
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
   94 CONTINUE
      CALL VELOT1(FELTBI)
      IF(.NOT.FELTBI) GO TO 25
      V=VV
      W=WV
      IF(ZDIMED) GO TO 96
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   96 CONTINUE
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
   25 IF(.NOT.FELT) GO TO 100
C
C******************** CORNER 3 *************************************
C
   20 IF (EM2 .LT. 0.0) GO TO 19
C
C     CORNER 3 FOR EM2 .GE. 0 ---PANEL TRAILING EDGE SWEPT BACK
C
      IF(XI.GE.XRBT(I)) GO TO 40
      ICR=3
      EML=EM2
      X=-XI+XLBT(I)
      Y=-YI+YLCT(I)
      Z=-ZI+ZLCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 102
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
  102 CONTINUE
      CALL VELOT1(FELTA)
      IF(.NOT.FELTA) GO TO 23
      V=VV
      W=WV
      IF(ZDIMED) GO TO 104
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
  104 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
   23 Y=YI+YLCT(I)
      YV=Y
      IF(ZDIMED) GO TO 106
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
  106 CONTINUE
      CALL VELOT1(FELTAI)
      IF(.NOT.FELTAI) GO TO 30
      V=VV
      W=WV
      IF(ZDIMED) GO TO 108
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
  108 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
      GO TO 30
C
C     CORNER 3 FOR EM2 .LT. 0 ---PANEL TRAILING EDGE SWEPT FORWARD
C
   19 IF(XI.GE.XLBT(I)) GO TO 40
      ICR=3
      EML=-EM2
      X=-XI+XLBT(I)
      Y=YI-YLCT(I)
      Z=-ZI+ZLCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 110
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
  110 CONTINUE
      CALL VELOT1(FELTA)
      IF(.NOT.FELTA) GO TO 26
      V=VV
      W=WV
      IF(ZDIMED) GO TO 112
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
  112 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
   26 Y=-(YI+YLCT(I))
      YV=Y
      IF(ZDIMED) GO TO 114
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
  114 CONTINUE
      CALL VELOT1(FELTAI)
      IF(.NOT.FELTAI) GO TO 21
      V=VV
      W=WV
      IF(ZDIMED) GO TO 116
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
  116 CONTINUE
      TU=TU-U
      TV=TV+V
```

Figure A-1(yy)

```
      TW=TW-W
      GO TO 21
C
C******************** CORNER 4 ************************************
C
C     CORNER 4 FOR EM2 .GE. 0
C
   30 CONTINUE
      ICR=4
      X=-XI+XRBT(I)
      Y=-YI+YRCT(I)
      Z=-ZI+ZRCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 120
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
  120 CONTINUE
      CALL VELOT1(FELTB)
      IF(.NOT.FELTB) GO TO 35
      V=VV
      W=WV
      IF(ZDIMED) GO TO 122
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
  122 CONTINUE
      TU=TU-U
      TV=TV+V
      TW=TW-W
   35 Y=YI+YRCT(I)
      YV=Y
      IF(ZDIMED) GO TO 124
      YV=ROTAC(Y,Z)
      ZV=ROTBC(Y,Z)
  124 CONTINUE
      CALL VELOT1(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      V=VV
      W=WV
      IF(ZDIMED) GO TO 126
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
  126 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      GO TO 40
C
C     CORNER 4 FOR EM2 .LT. 0
C
   21 CONTINUE
      ICR=4
      X=-XI+XRBT(I)
      Y=YI-YRCT(I)
      Z=-ZI+ZRCT(I)
      ZV=Z
      YV=Y
      IF(ZDIMED) GO TO 130
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
  130 CONTINUE
      CALL VELOT1(FELTB)
      IF(.NOT.FELTB) GO TO 28
      V=VV
      W=WV
      IF(ZDIMED) GO TO 132
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
  132 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
   28 Y=-(YI+YRCT(I))
      YV=Y
      IF(ZDIMED) GO TO 134
      YV=ROTA(Y,Z)
      ZV=ROTB(Y,Z)
  134 CONTINUE
      CALL VELOT1(FELTBI)
      IF(.NOT.FELTBI) GO TO 40
      V=VV
      W=WV
      IF(ZDIMED) GO TO 136
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
  136 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
C
C
C
C     CALCULATION OF PERTURBATION VELOCITIES
C     SIGN CHANGES NECESSARY TO ACCOMMODATE USE OF SUBROUTINE
C     VELOTH WITH DIFFERENT COORDINATE SYSTEM
C
   40 CONTINUE
      UP=UP-TU*DZDX(I)
      VP=VP+TV*DZDX(I)
      WP=WP-TW*DZDX(I)
  100 CONTINUE
      RETURN
      END


      SUBROUTINE WDYBDY
C
C
C     SUBROUTINE TO INPUT AND OUTPUT FUSELAGE DATA
C        AND GENERATE SOURCE PANEL SOLUTION FOR ARBITRARY BODY
C        USING WOODWARD SUPERSONIC SOURCE PANEL METHOD
C
C
      COMMON /BIPYZ / XBIP,NYZBIP,YCB(33),ZCB(33)
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /CONSTS/ PI,PI2,DTOR,RTOD,FOURPI
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITM,IDEL,NTAP7
     *    ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *    ,NAG,NAP,NAV,NAS,NASHK,NAFLO,IAO,IDO,ISKO,IZ2(4),NRING,IROW(50)
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETAO,BETASQ,FMCHSQ
      COMMON /FSGEOM/ FRMAX,SSPAN,    FLTHC,BODYPL
      COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101)
      COMMON /INDEX / NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,NP
     1    ,NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1    ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON /WGEOM / XRWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     ,       PSIWTE(20),Y(20),PHID(20),ZDIMED,WICR
C
  701 FORMAT(10I5)
  706 FORMAT(8F10.1)
  708 FORMAT(10X,19HFUSELAGE INPUT DATA
     *    /15X,31HWOODWARD SOURCE PANELING METHOD/)
  709 FORMAT(15X,17HFUSELAGE LENGTH =,13X,F10.5,5H FEET/15X
     1    ,16HMAXIMUM RADIUS =,14X,F10.5,5H FEET/15X
     2    ,22HNOSE TIP SHOCK ANGLE =,8X,F10.5,8H DEGREES/15X
     3    ,27HINTERFERENCE SHELL LENGTH =,3X,F10.5,5H FEET/)
  732 FORMAT(1H1,10X,32HFUSELAGE INTERFERENCE SHELL DATA
     *    //14X,I3,44H RINGS OF U-VELOCITY PANELS ARE TO BE LAID OUT
     1    39H ON THE FUSELAGE FROM XW= 0 FEET TO XW=,F10.5,5H FEET
     2    /20X,9HTHERE ARE,I3,54H PANELS IN EACH RING ON THE LEFT HALF OF
```

Figure A-1(zz)

```
    3 THE FUSELAGE,/20X,25HTOTAL NUMBER OF PANELS IS ,I4)
  734 FORMAT(/15X,25HFUSELAGE SHOCK WAVE SHAPE)
C
C     READ IN AND WRITE FUSELAGE AND INTERFERENCE SHELL GEOMETRY
C
      READ (5,706) FLTHC,FRMAX,FTHSHK,BODYPL
      BODYPL = ABS(BODYPL)
      WRITE (6,708)
      WRITE (6,709) FLTHC,FRMAX,FTHSHK,BODYPL
      READ (5,701) NCWB
C
C  READ WDYBDY GEOMETRY
C
      CALL GEOM
C
C  SAVE INTERFERENCE PANEL LAYOUT SECTION
C
      CALL YZBIP
      NBD = NYZBIP-1
      NBDCR2 = 0
      NBDCR1 = NBD
      BB = -BODYPL
      REFL = AMAX1(REFL,FLTHC)
      NBIP = NBD*NCWB
      NBDCR1 = 0
      NBDCR2 = NBD
      WRITE (6,732) NCWB,BB,NBD,NBIP
      WRITE (6,150)  XBIP
      DO 140 I=1,NYZBIP
140   WRITE(6,160) I,YCB(I),ZCB(I)
150   FORMAT(///10X,43HY-Z COORDINATES OF BODY INTERERENCE PANELS
     *   7H XBIP =,G12.5,//19X,1HI,8X,3HYCB,9X,3HZCB)
160   FORMAT(15X,I5,2F12.5)
C
C  COMPUTE SOURCE STRENGTHS
C
      XMACH = FMACH
      ALPHAC = ALFACR*RTOD
      ALPHA = ALFACR*RTOD
      PHIR = 0.0
      IUVW = 1
      CALL VELCMP (1)
      CALL SOLVE  (1)
C
C     CALCULATE FUSELAGE SHOCK WAVE SHAPE
C
C             IF SHOCK REQUIRED IS GREATER THAN 90 DEGREES, USE INPUT
C             OPTION TO SPECIFY VALUES
      PHISHK = 90.0
      WRITE (6,734)
      CALL BSHOCK(FRMAX,PHISHK,FTHSHK)
C
C  IF INLET EXISTS, COMPUTE SPECIAL SHOCK SHAPE FOR RAMP INLETS
C
      CALL INLSHK(PHISHK)
C
C  SAVE SOLUTION ON TAPE10  AS TEMPORARY FILE  IF STORES EXIST
C
      CALL FRSTRT (10,1,LASTA)
      RETURN
      END



      SUBROUTINE WITHIN
C
C     SUBROUTINE TO INPUT WING THICKNESS DATA
C
      COMMON/THKDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     1   XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2   THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     3   YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4   PSPSTE(20),SNPHS(400),CSPHS(400)
C
  701 FORMAT (10I5)
  702 FORMAT (8F10.0)
  703 FORMAT(//1X,I4,55H THICKNESS PANELS ARE TO BE LAID OUT ON EACH WIN
     1G PANEL/20X,I3,20H CHORDWISE ROWS WITH,I3,12H IN EACH ROW/
     2  22X,29HWING THICKNESS OPTION, NUNIS=,I2//)
  704 FORMAT(//5X,90H***** SLOPE OF WING THICKNESS ENVELOPE AT LEADING E
     1DGE NOT POSITIVE FOR SOME CHORDWISE ROW,15X,12H** WITHIN **)
  705 FORMAT(I5,4F10.0)
  706 FORMAT(15X,59HSPANWISE LOCATIONS OF PANEL SIDE EDGES AND SWEEP ANG
     1LES AND/20X,43HDIHEDRAL ANGLE OF WING SECTION TO THE RIGHT//
     2  24X,39HSPANWISE   LE SWEEP  TE SWEEP  DIHEDRAL/
     3  20X,1HI,5X,8HLOCATION/28X,4HFEET,5X,7HDEGREES,3X,
     4  7HDEGREES,3X,7HDEGREES)
  707 FORMAT(18X,I3,4X,4F10.5)
C
      READ(5,701) NCWS,MSWS,NUNIS
      MS=NCWS*MSWS
      WRITE (6,703) MS,MSWS,NCWS,NUNIS
      MS1=MSWS+1
      DO 1 J=1,MS1
    1 READ(5,705) I,YS(I),PSWSLE(I),PSWSTE(I),PHIS(I)
      WRITE(6,706)
      DO 2 J=1,MS1
    2 WRITE(6,707) J,YS(J),PSWSLE(J),PSWSTE(J),PHIS(J)
      IF (NUNIS .NE. 0) GO TO 5
      MN=0
      DO 3 JNW=1,MS,NCWS
      MN=MN+NCWS
    3 READ (5,702) (THETAL(J),J=JNW,MN)
      GO TO 10
    5 READ (5,702) (THETAL(J),J=1,NCWS)
      DO 6 J=2,MSWS
      JJ=(J-1)*NCWS
      DO 6 K=1,NCWS
      KK=JJ+K
    6 THETAL(KK)=THETAL(K)
C
C     CHECK FOR ZERO OR NEGATIVE LEADING EDGE SLOPE
C
   10 DO 15 JLE=1,MS,NCWS
      IF (THETAL(JLE).LE.0.0) GO TO 20
   15 CONTINUE
      RETURN
   20 WRITE (6,704)
      STOP
      END



      SUBROUTINE WLYOUT
C
C     SUBROUTINE TO LAYOUT CONSTANT U-VELOCITY PANELS ON THE WING
C
      COMMON/PPGEOM/SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),SNG
      COMMON/CONFIG/NFU,NPY,NSTRS,LVSWP,NRACK
      COMMON/CONSTS/PI,PI2,DTOR,RTOD,FOURPI
      COMMON/INDEX/NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP,MP,
     1   NCP,MSP,N1P,N2,N2P,NPTOT
      COMMON /WGEOM / XRWOC,ZRWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1     PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR
      COMMON/WPGEOM/XRF(200),XRB(200),XLF(200),XLB(200),YRC(200),
     1   YLC(200),XCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),SWPPTE(200),
     2      SNPHI(200),CSPHI(200)
C
      DATA FAC/0.95/
C
C     LAY OUT ELEMENTAL PANELS ON LEFT WING PANEL
C
      LVSWP=0
      IF(MSWP.EQ.2) GO TO 15
      DO 10 I=3,MSWP
      IM=I-1
```

Figure A-1(aaa)

```
10 IF(PSIWLE(I).NE.PSIWLE(IM).OR.PSIWTE(I).NE.PSIWTE(IM)) LVSWP=1
15 SLPWLE=TAN(PSIWLE(MSWP)*DTOR)
   SLPWTE=TAN(PSIWTE(MSWP)*DTOR)
   SLPDIF=SLPWLE-SLPWTE
   CSIDEP=CRW
   WLEX=0.0
   ZLER=0.0
   ANCW=NCW
C
C     LOOP OVER WING CHORDWISE ROWS
C
   DO 140 I=2,MSWP
   IM=I-1
   DY=Y(I)-Y(IM)
   IF(LVSWP.EQ.0) GO TO 40
   SLPWLE=TAN(PSIWLE(I)*DTOR)
   SLPWTE=TAN(PSIWTE(I)*DTOR)
   SLPDIF=SLPWLE-SLPWTE
40 CSIDE=CSIDEP-DY*SLPDIF
   IF(I.EQ.2) GO TO 45
   JLE=(I-3)*NCW+1
   WLEX=XLF(JLE)
   ZLER=ZLC(JLE)
45 CONTINUE
   CSP=COS(PHID(I)*DTOR)
   SNP=SIN(PHID(I)*DTOR)
   TNPHI=SNP/CSP
C
C     LOOP OVER ELEMENTAL PANELS IN THIS CHORDWISE ROW
C
   DO 130 K=1,NCW
   J=(I-2)*NCW+K
   AK=K
   AKM=K-1
   SWPPLE(J)=SLPWLE-AKM*SLPDIF/ANCW
   SWPPTE(J)=SLPWLE-AK*SLPDIF/ANCW
   CSPHI(J)=CSP
   SNPHI(J)=SNP
C
C     CORNER POINT COORDINATES
C
   XRF(J)=-AKM*CSIDEP/ANCW+WLEX
   XRB(J)=XRF(J)-CSIDEP/ANCW
   XLF(J)= -AKM*CSIDE/ANCW+DY*SLPWLE+WLEX
   XLB(J)=XLF(J)-CSIDE/ANCW
   YRC(J)=Y(IM)
   YLC(J)=Y(I)
   ZRC(J)=ZLER
   ZLC(J)=ZLER+DY*TNPHI
C
C     Y-COORDINATES OF CONTROL POINT AND PANEL CENTROID
C
   IF(K.GT.1) GO TO 50
   A1=XLB(J)-XLF(J)
   A2=XRB(J)-XRF(J)
   H=-DY
   YBAR=(2.0*A1+A2)*H/(3.0*(A1+A2))
50 CONTINUE
   YCPT(J)=Y(IM)-YBAR
C
C     X-COORDINATES OF CONTROL POINT
C
   XPCLE=XRF(J)-YBAR*SWPPLE(J)
   XPCTE=XRB(J)-YBAR*SWPPTE(J)
   PNLC=XPCLE-XPCTE
   XCPT(J)=XPCLE-FAC*PNLC
C
C     Z-COORDINATES OF CONTROL POINT
C
   ZCPT(J)=ZLER-YBAR*TNPHI
130 CONTINUE
   CSIDEP=CSIDE
140 CONTINUE
C
   RETURN
   END
```

```
      SUBROUTINE WRFILE
C
C     THIS SUBROUTINE WRITES ALL OF THE DATA WHICH IS READ INTO OR
C          COMPUTED IN PROGRAM 1 AND IS USED IN THE TRAJECTORY PROGRAM,
C          PROGRAM 2, ON TAPE 12
C
C     COMMON STATEMENTS
C
      COMMON /BASFSG/ FBASE,FSUMR,FSUMRD,RBASE,RSUMR,RSUMRD,SBASE(7),
     1 SSUMR(7),SSUMRD(7)
      COMMON /CONFIG/NF[illegible],NPY,NSTRS,LVSWP,NHAC[illegible]
      COMMON /FLOW/ALFACR,GAMR,FMACH,RHO[illegible],INF,BETA,BETASQ,FMCHS,
      COMMON /FSGEOM/FAMAX,SSPAN,  FLTHC,HOLTR,
      COMMON /FSHOCK/ NFSHR,FASHR(50),FRSHR(50),F[illegible](10)
      COMMON /FSOR/FXL(101),FSS(100),FCS(100),NFSOR
      COMMON/NUIN[illegible]/N[illegible],NR[illegible],NRW,NRP,NRH,NPTW,NPTP,NPTH,NRTOT,NPTT,
     1      NRWS,NRPS,NPTWS,NPTPS,NRTOTS,NPTTS,NRWP,NPTWP,
     2      NCW1,NCP1,NCWB1,NCWS1,NCPS1
      COMMON/PYGEOM/Z(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
     1      SLLE,PSLPDF,CENTER,ZPL,LVSPP
      COMMON /RRGEOM/ RRMAX,RLTHC,XRWOC,YRO,ZRO,NRPOLY,RXEND(7),
     1        RCOEF(7,7),XHR[illegible],YHRO,ZHRO,RIHCH,SRIHCH,CHIRCH
      COMMON /RSHOCK/ NRSHR,RXSHR(50),RRSHR(50),R[illegible](10)
      COMMON /RSOR/ RXL(101),RSS(100),RCS(101),NRSOR
      COMMON/SHPDAT/NFPOLY,FXEND(7),FCOEF(7,7)
      COMMON /SSHOCK/ NSSHR(7),SXSHR(50,7),SRSHR(50,7),SCH[illegible](10,7)
      COMMON /SSOR/ SXL(101,7),SSS(100,7),SCS(100,7),N[illegible](7)
      COMMON /STGEOM/SHMAX(7),SLTHC(7),XWS[illegible](7),YWS(7),ZWS(7),
     1       NSPOLY(7),SXEND(7,7),SCOEF(7,7,7),XHS[illegible](7),YHSO(7),ZHS[illegible](7),
     2       SIRCR(7),NUMSTR(7),SSIBCR(7),CSIHCH(7),SPHIRR(7),NSHAPE(7)
     3       ,NSHPT,MSHAPE(7)
      COMMON/THRDAT/NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     1    XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     2    THETAL(400),THETPL(200),SLLET(400),SLTET(400),CZLT(400),
     3      YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     4      PSPSTE(20),SNPHS(400),CSPHS(400)
      COMMON /VELAR /X,DUMY,[illegible],FM,TLRNC,TIRY,YYPNL
      COMMON /WGEOM / XRWOC,ZRWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
     1      PSIWTE(20),Y(20),PHID(20),ZOIHED,WICP
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
      COMMON            XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
     1      SWP(500),SNAP(40),CSHP(40),THTHP(40),DPNET(500),
     2      XPTS(1000),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
     3      SWPS(1000),THTNET(1000)
      DIMENSION       A(12000)
      EQUIVALENCE (XPT(1),A(1))
      REWIND 10
C
C     WRITE INDICES
C
      WRITE(12) NFU,NPY,NSTRS,NRW,NRP,NRH,NPTW,NRWS,NRPS,NPTWS,NRWP,
     1     NPTWP,NCW1,NCWS,NCWS1,NRTOT,NPTT,NRTOTS,NPTTS,NTHP,NWA[illegible]
C
C     WRITE CONSTANTS
C
      WRITE (12) ALFACR,FMACH,TLRNC,XRWOC,ZRWO,CRW,ZOIHED,BETA,BETASQ
C
C     WRITE U VELOCITY PANEL DATA
C
      WRITE(12)(XPT(I),DPNET(I),I=1,NPTT)
      WRITE(12)(SWP(I),I=1,NPTWP)
      WRITE(12)(YPT(I),ZPT(I),I=1,NRTOT)
      WRITE(12)(SPHI(I),CPHI(I),I=1,NRW)
C
C     WRITE THICKNESS PANEL DATA
C
      WRITE(12)(XPTS(I),SWPS(I),THTNET(I),I=1,NPTTS)
      WRITE(12)(YPTS(I),ZPTS(I),I=1,NRTOTS)
```

Figure A-1(bbb)

```
      WRITE(12)(SPHS(I),CPHS(I),I=1,NRWS)
      WRITE(12)(DZDX(I),I=1,NTHP)
C
C     WRITE FUSELAGE DATA
C
      IF (NFU.EQ.0) GO TO 15
      WRITE(12) NCWB1,(SNAP(I),CSBP(I),THTBP(I),I=1,NRB)
C
C              WRITE ARRAY INFORMATION FOR LINE-SOURCE METHOD
      IF (NFU.NE.1) GO TO 10
      WRITE(12)FRMAX,FLTHC,NFPOLY,(FXEND(I),I=1,NFPOLY),
     1    ((FCOEF(I,J),J=1,7),I=1,NFPOLY)
      WRITE (12) NFSOR,(FXL(I),FSS(I),FDS(I),FDRDX(I),I=1,NFSOR)
     1 ,FBASE,FSUMK,FSUMKD,FXSHLD
      WRITE (12) NFSHK,(FXSHK(I),FRSHK(I),I=1,NFSHK)
      GO TO 15
10    IF (NFU.NE.2) GO TO 15
C
C              WRITE ARRAY INFORMATION FOR SOURCE PANEL METHOD
      CALL FRSTRT (10,2,LASTA)
      CALL FRSTRT (12,1,LASTA)
C
C     WRITE PYLON DATA
C
15    IF (NPY.EQ.0) GO TO 20
      WRITE(12) YPL,CENTER,NCPI,NCPSI
C
C     WRITE RACK DATA
C
      IF (NRACK.EQ.0) GO TO 20
      WRITE (12) RLTHC,RRMAX,RIBCR,CRIBCR,SRIBCR,YBRO,XBRO,ZBRO,NRSOR,
     1 NRSHK,RBASE,RSUMK,RSUMKO,RXSHLO
      WRITE (12) (RXL(I),RSS(I),RDS(I),RDRDX(I),I=1,NRSOR)
      WRITE (12) (RXSHK(I),RRSHK(I),I=1,NRSHK)
C
C     WRITE STORE DATA
C
   20 IF (NSTRS.EQ.0) RETURN
      WRITE (12) (SLTHC(I),SRMAX(I),SIBCR(I),CSIBCR(I),SSIBCR(I),
     1   YBSO(I),XBSO(I),ZBSO(I),NUMSTR(I),NSSOR(I),NSSHK(I),NSHAPE(I),
     2   SBASE(I),SSUMK(I),SSUMKD(I),SXSHLD(I),SPHIRR(I),I=1,NSTRS)
      WRITE (12) NSHPT,(NSHAPE(I),I=1,NSHPT)
      DO 30 J=1,NSTRS
      IF (NSHAPE(J).GT.50) GO TO 30
C
C              SAVE STORE INFORMATION FOR LINE SOURCE METHOD
      MSOR=NSSOR(J)
      WRITE (12) (SXL(I,J),SSS(I,J),SDS(I,J),SDRDX(I,J),I=1,MSOR)
      MSSHK=NSSHK(J)
      WRITE (12) (SXSHK(I,J),SRSHK(I,J),I=1,MSSHK)
30    CONTINUE
C
C              SAVE STORE INFORMATION FOR SOURCE PANEL METHOD
      DO 40 ISHP=1,NSHPT
      IF (MSHAPE(ISHP).LE.50) GO TO 40
      CALL FRSTRT (10,4,LASTA)
      CALL FRSTRT (12,3,LASTA)
40    CONTINUE
      RETURN
      END


      SUBROUTINE YZBIP
C
C ROUTINE TO DEFINE Y-Z SECTION GEOMETRY FOR BODY INTERFERENCE
C PANEL LAYOUT IN BLYOT2.  PROGRAM SCANS FROM MERIDIONAL INTERPOLATION
C TO FIND DATA AT AXIAL LOCATION XBIP.
C
      COMMON /POPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XMLE
     3   ,REFA,REFO,REFL,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /FUSEUM / [illegible]
     *   ,FUSAZ(5)),[illegible],PHI(33)
      COMMON /DIMENS/ [illegible],NAFH,NAFR,MAXNA,MAXPH,NATOT
     *   ,[illegible]
     *   ,IXZSYM,[illegible],NTART
     *   ,IAN,IAN,[illegible]
     *   ,NAG,NAP,NAV,NAS,NASH,[illegible]
      COMMON /ZBIPYZ / XBIP,NYZBIP,YCB(33),ZCB(33)
      COMMON          A(12000)
      REWIND 8
      IXB = 1
      DO 120 IFU=1,NFUS
      NYZ = NFORX(IFU)*KRADX(IFU)
      IF (NFUS.GT.1) CALL IOREAD (8,A(IYB),2*NYZ)
      IZB = IYB+NYZ
      CALL YZBIP2 (XFUS(IXB),A(IYB),A(IZB),KRADX(IFU),NFORX(IFU))
      IXB = IXB+NFORX(IFU)
120   CONTINUE
      REWIND 8
      RETURN
      END


      SUBROUTINE YZBIP2(XB,YB,ZB,KRAD,NFUSOR)
C
C ROUTINE TO PERFORM THE INTERPOLATION IN THE BODY GEOMETRY FOR
C THE BODY INTERFERENCE CROSS SECTION DATA.
C
      DIMENSION XB(NFUSOR),YB(NFUSOR,KRAD),ZB(NFUSOR,KRAD)
      COMMON /BIPYZ / XBIP,NYZBIP,YCB(33),ZCB(33)
C
      XIN(X1,Y1,X2,Y2,Y)=X1+(X2-X1)*(Y-Y1)/(Y2-Y1)
      IF (XBIP.LT.XB(1) .OR. XBIP.GT.XB(NFUSOR))    RETURN
C
C SECTION LIES WITHIN THIS SEGMENT -------------------------------------
C INTERPOLATE FOR SECTION AXIAL LOCATION
C
      DO 100 J=2,NFUSOR
      IF (XBIP.LT.XB(J)) GO TO 110
100   CONTINUE
110   J = NFUSOR
      JM1 = J-1
      NYZBIP = KRAD
      DO 120 K=1,KRAD
      YCB(K) = XIN(YB(JM1,K),XB(JM1),YB(J,K),XB(J),XBIP)
      ZCB(K) = XIN(ZB(JM1,K),XB(JM1),ZB(J,K),XB(J),XBIP)
120   CONTINUE
      RETURN
      END
```

Figure A-1(ccc)

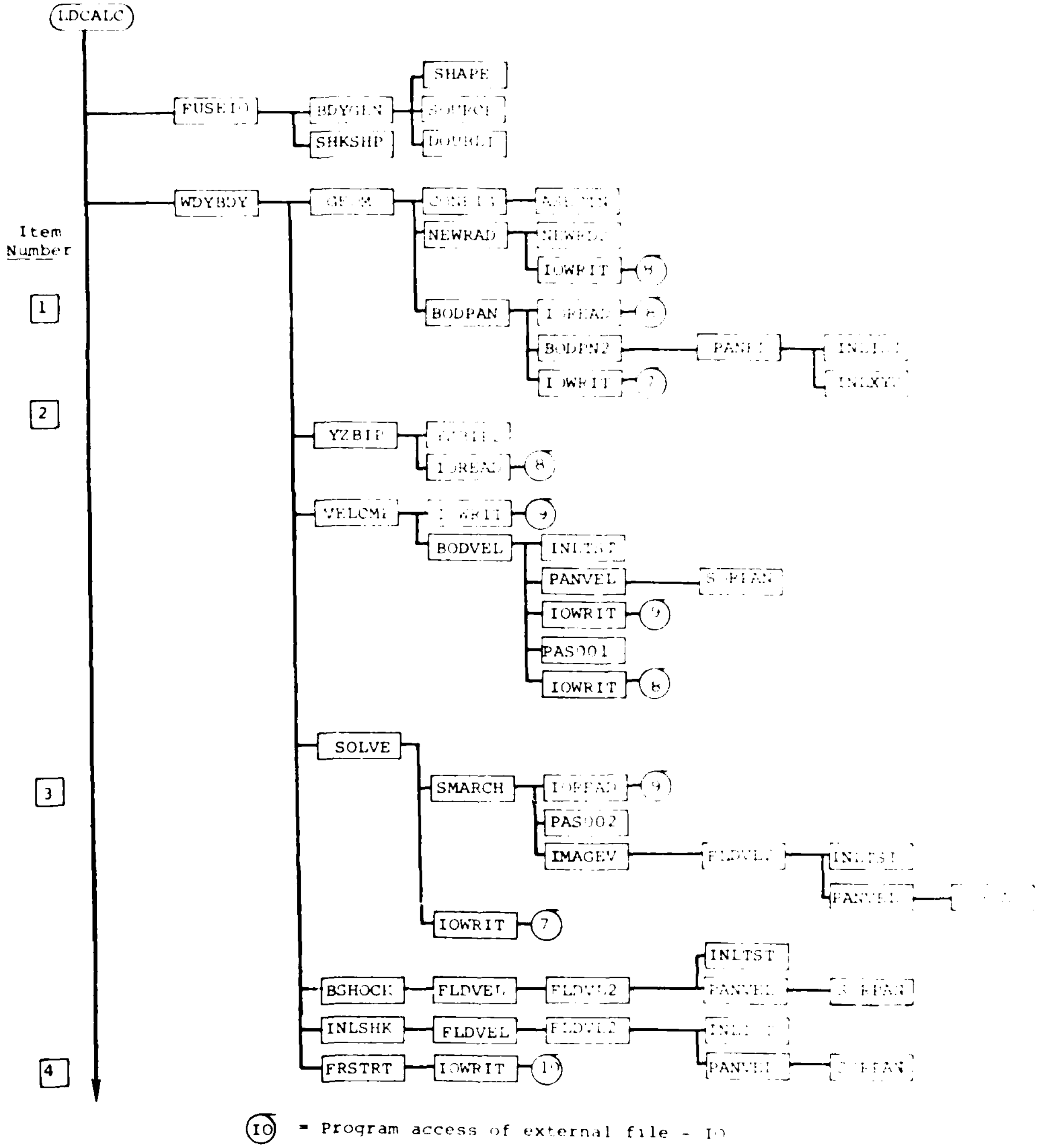


Figure A-2.- General flow chart of Subroutine Calls in Program 1.

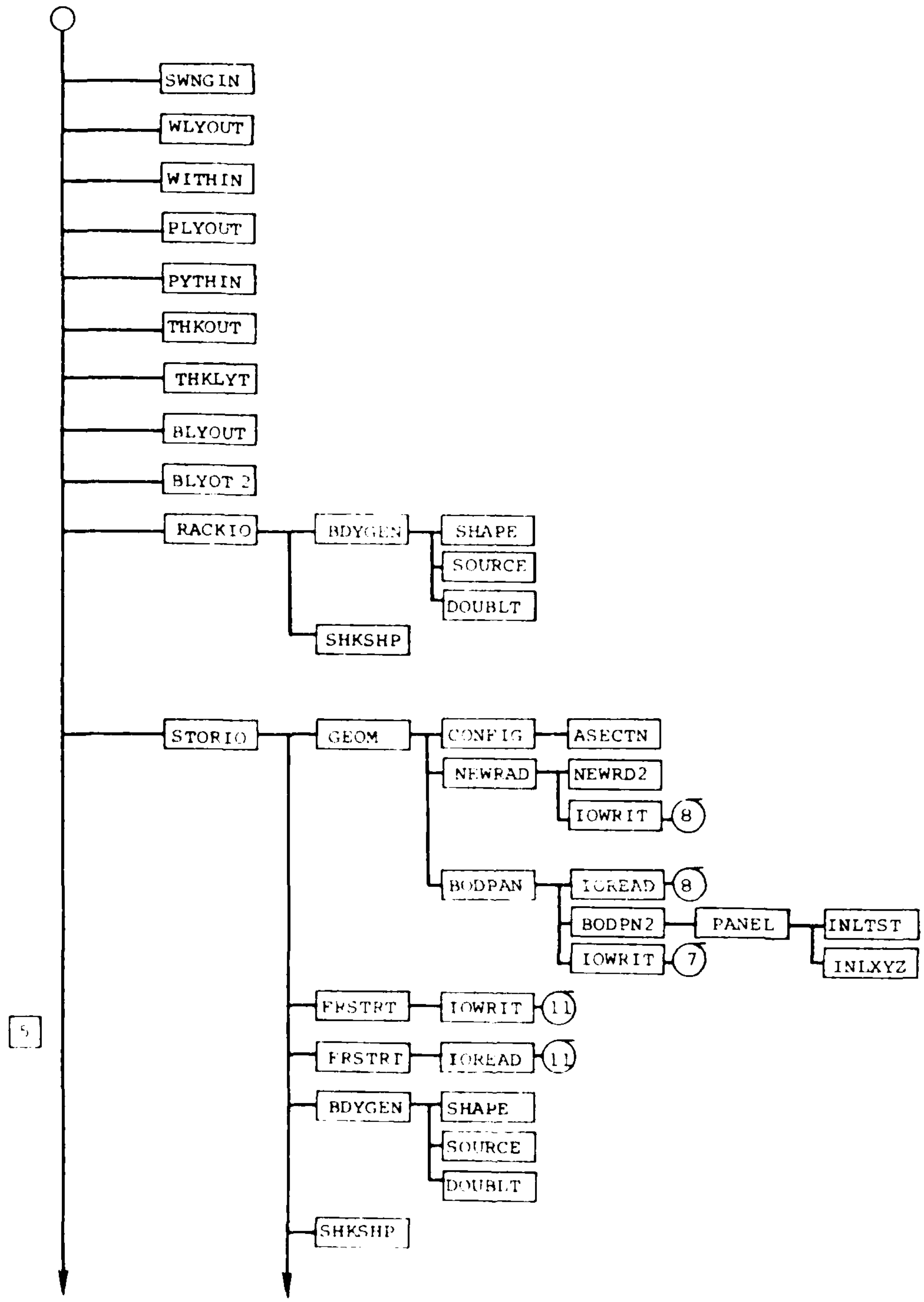


Figure A-2.- Continued.

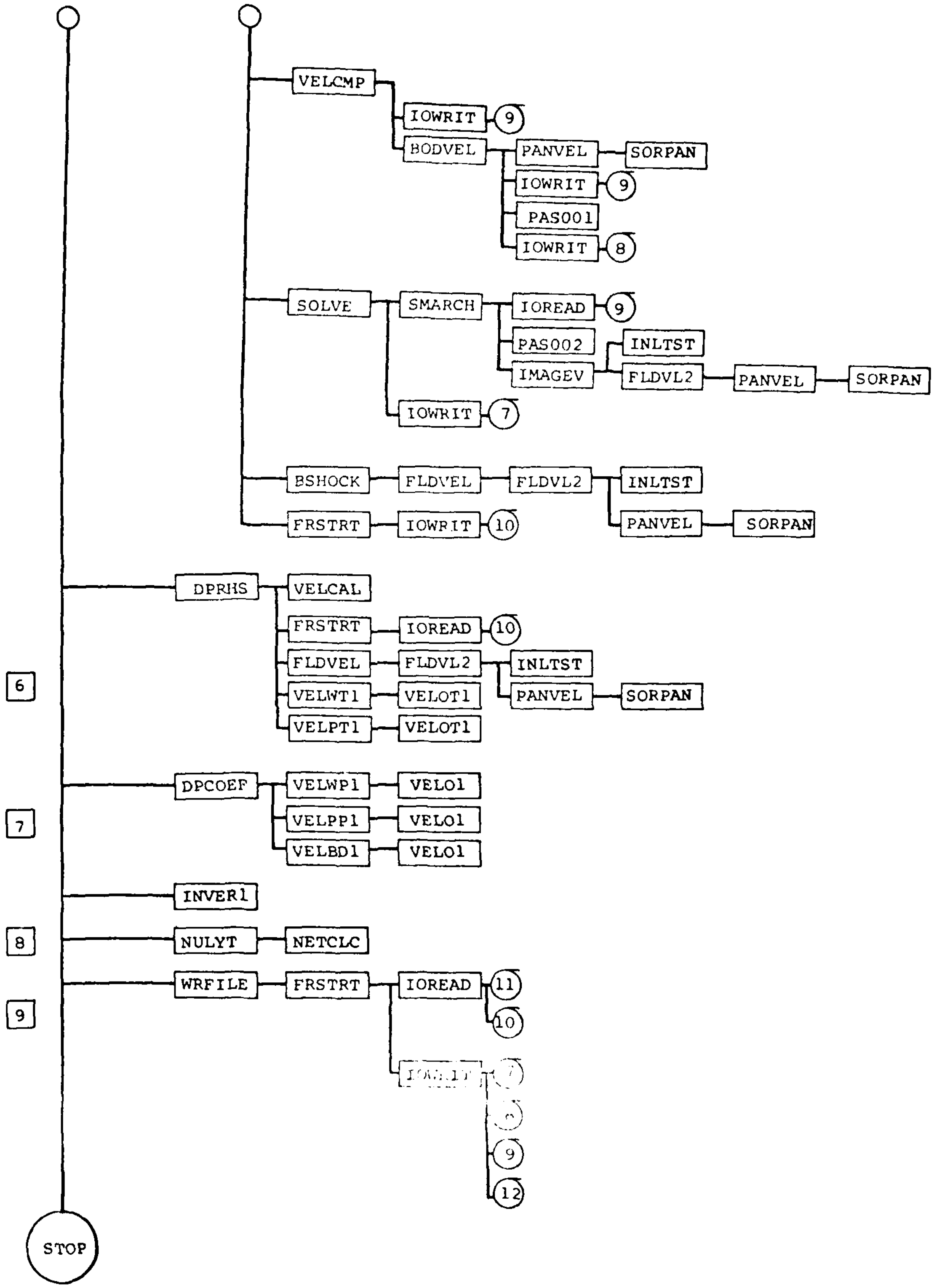
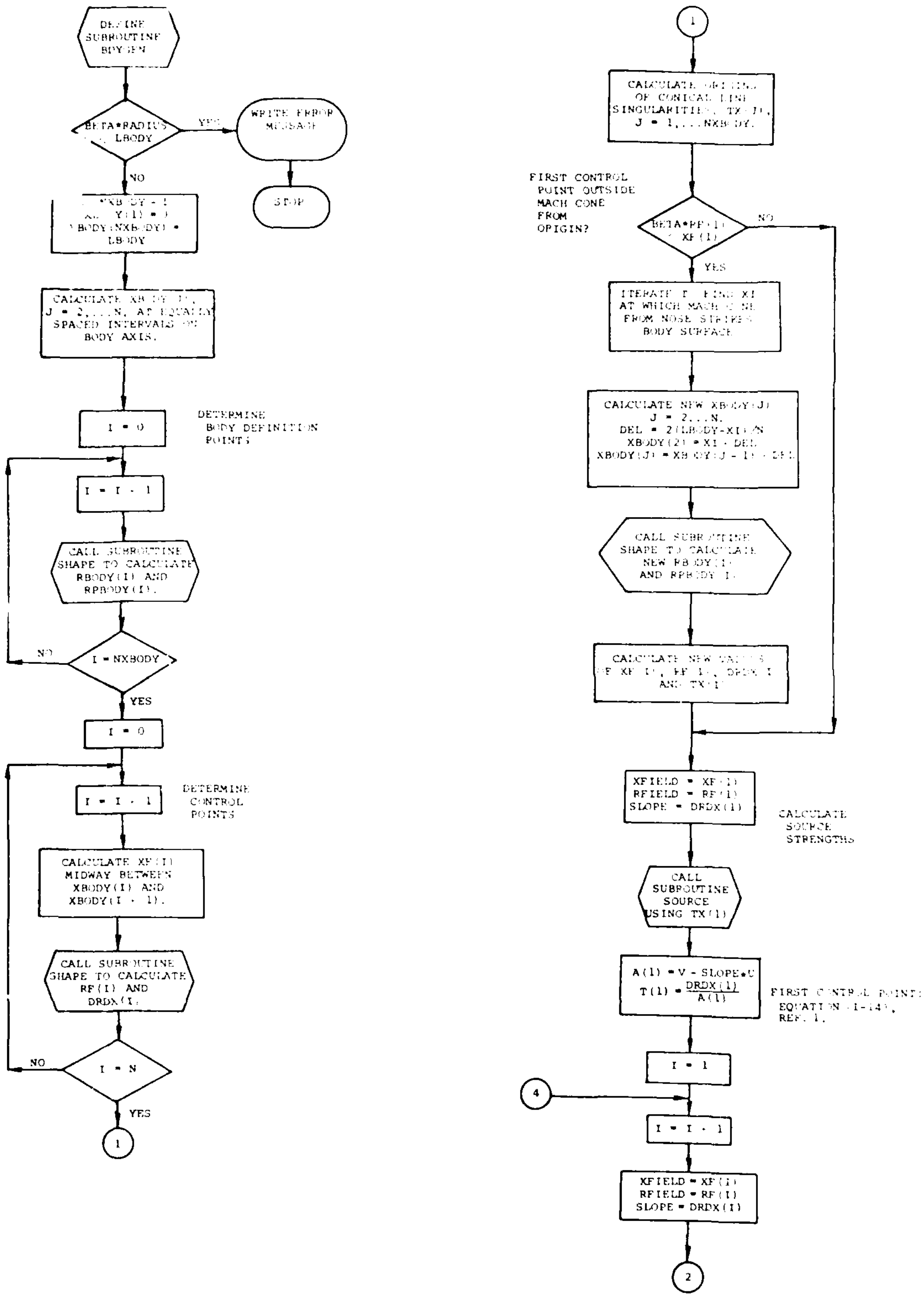
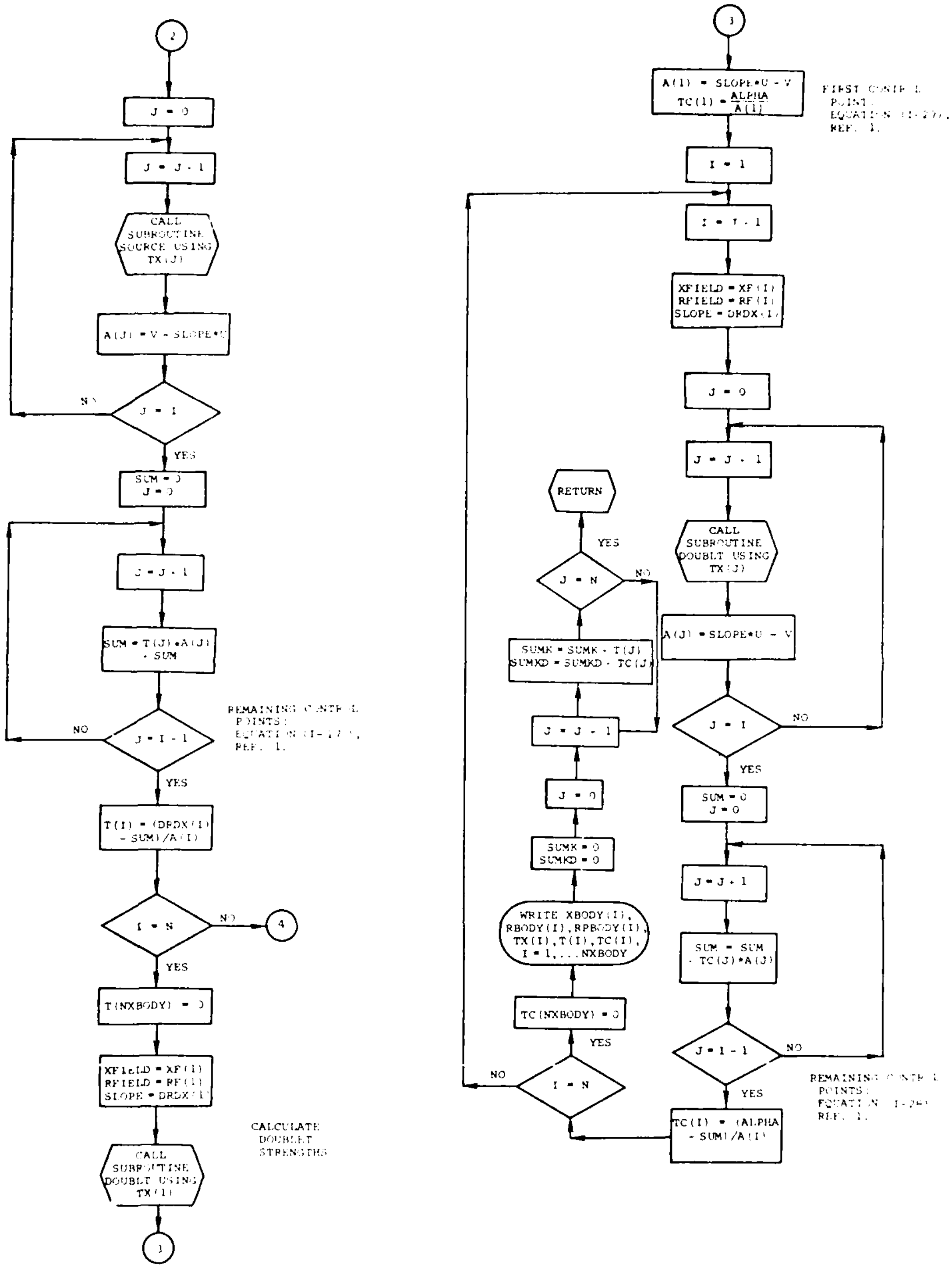


Figure A-2.- Concluded.



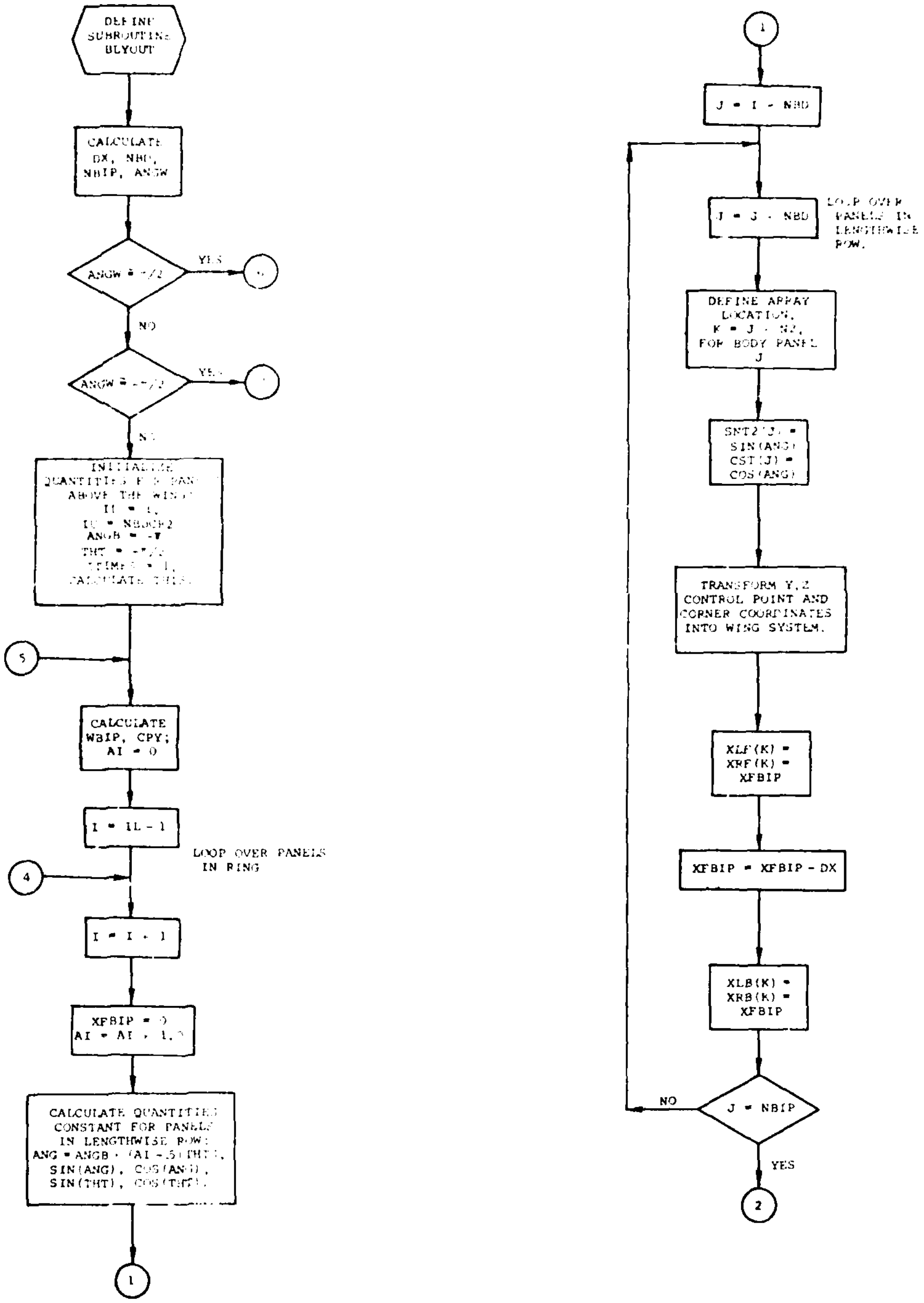
(a)

Figure A-3.- Flow chart of subroutine BDYGEN.



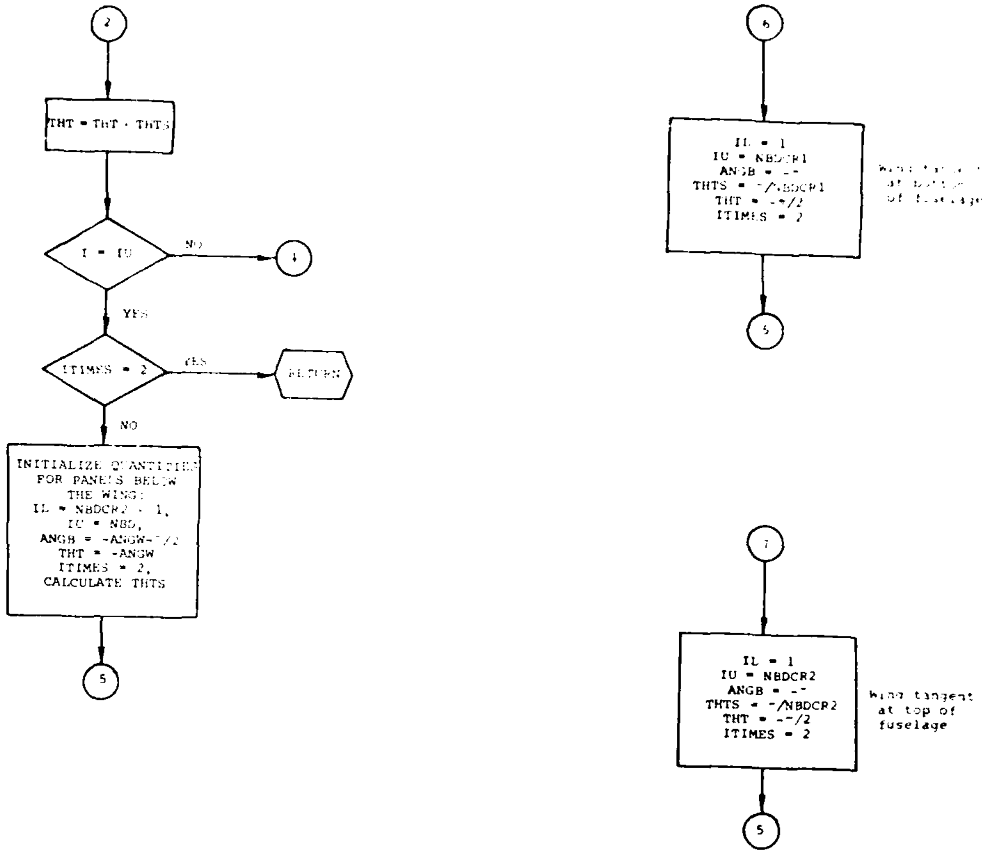
(b)

Figure A-3.- Concluded.



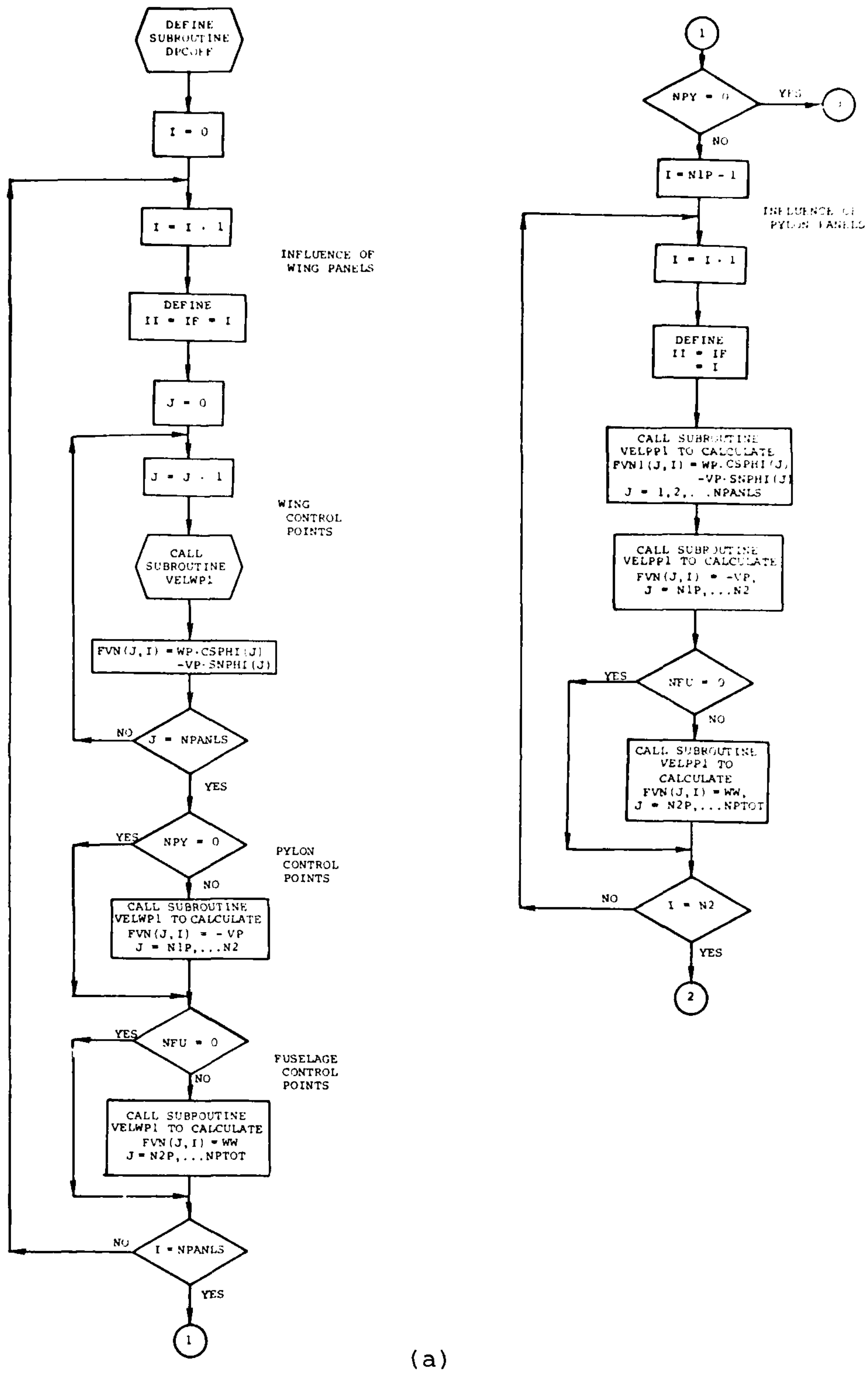
(a)

Figure A-4.- Flow chart of subroutine BLYOUT.



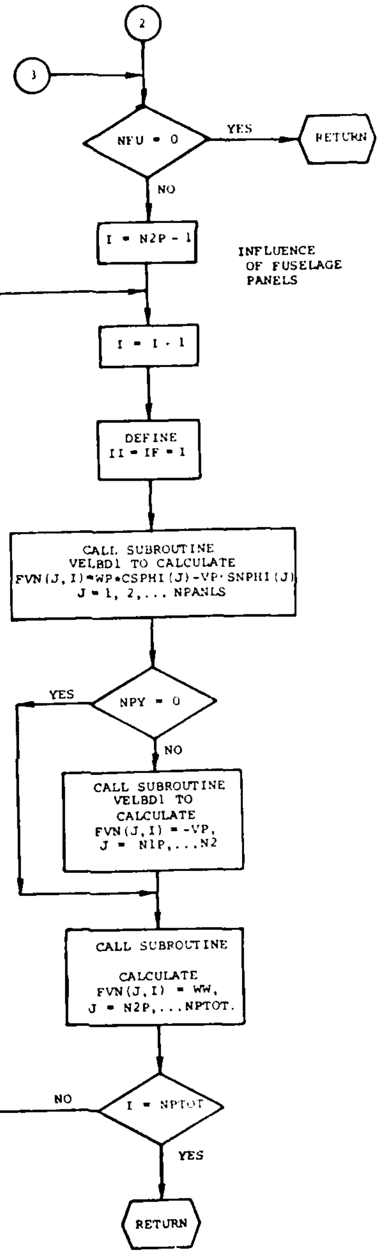
(b)

Figure A-4.- Concluded.



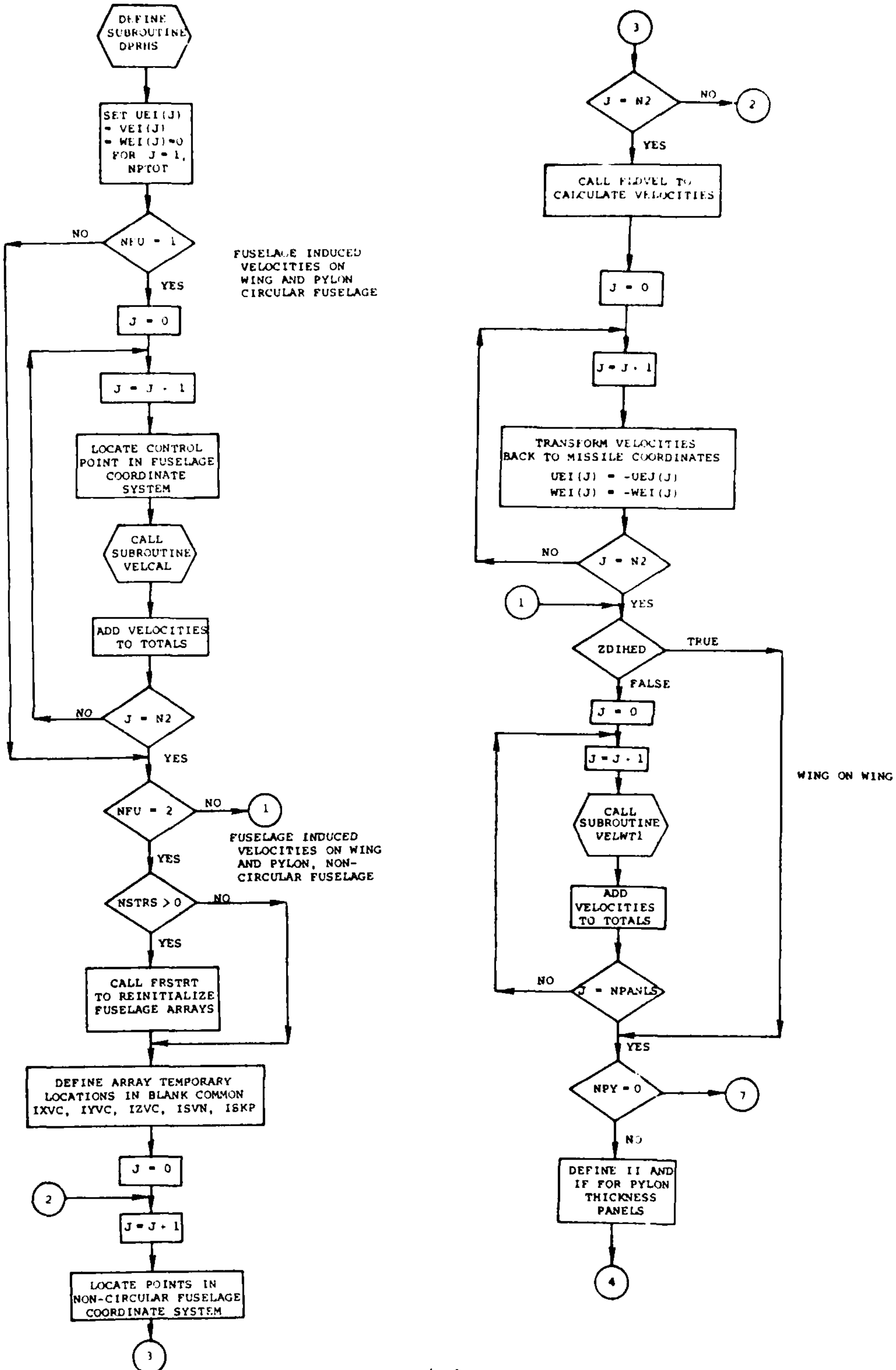
(a)

Figure A-5.- Flow chart of subroutine DPCOEF.

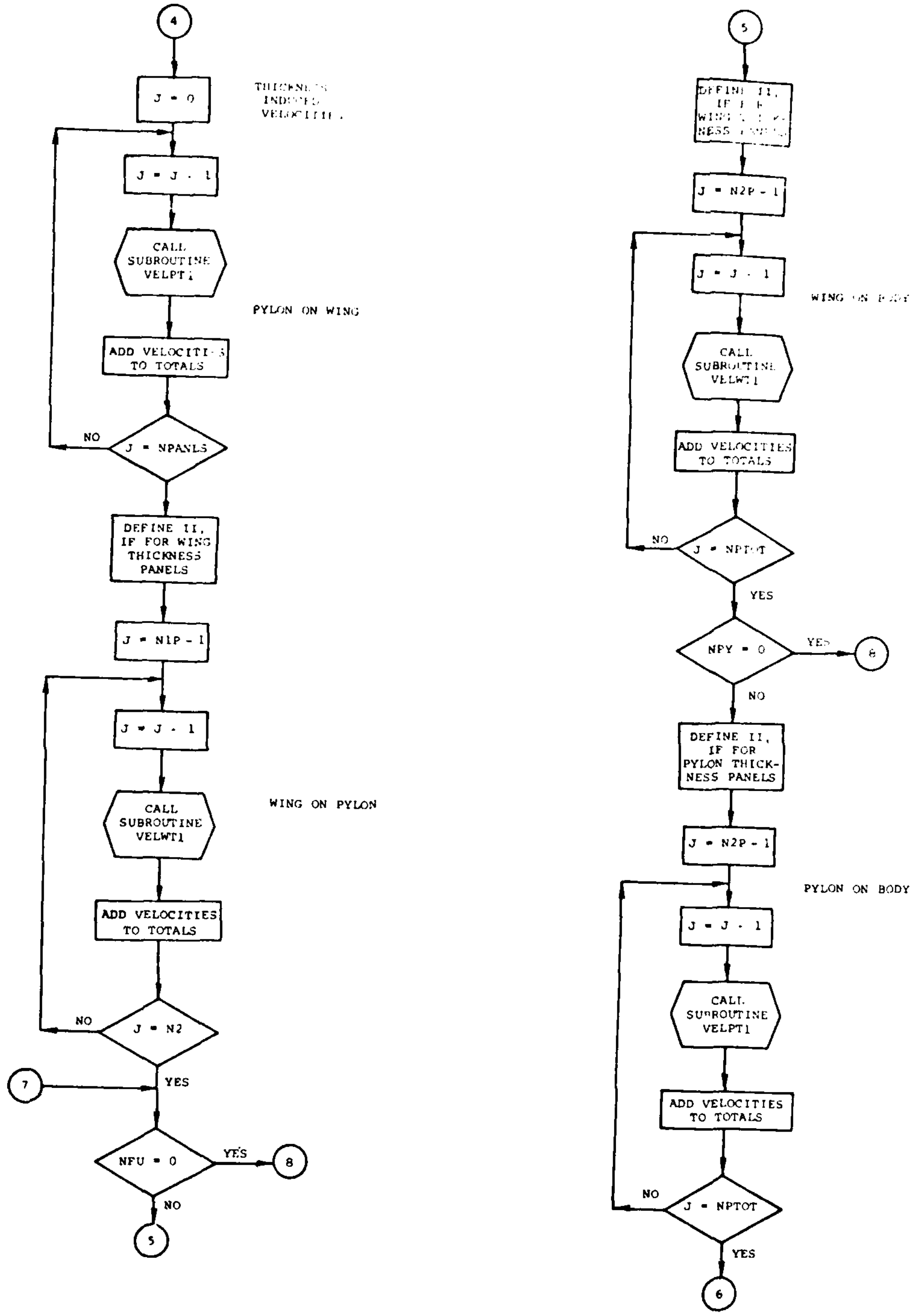


(b)

Figure A-5.- Concluded.

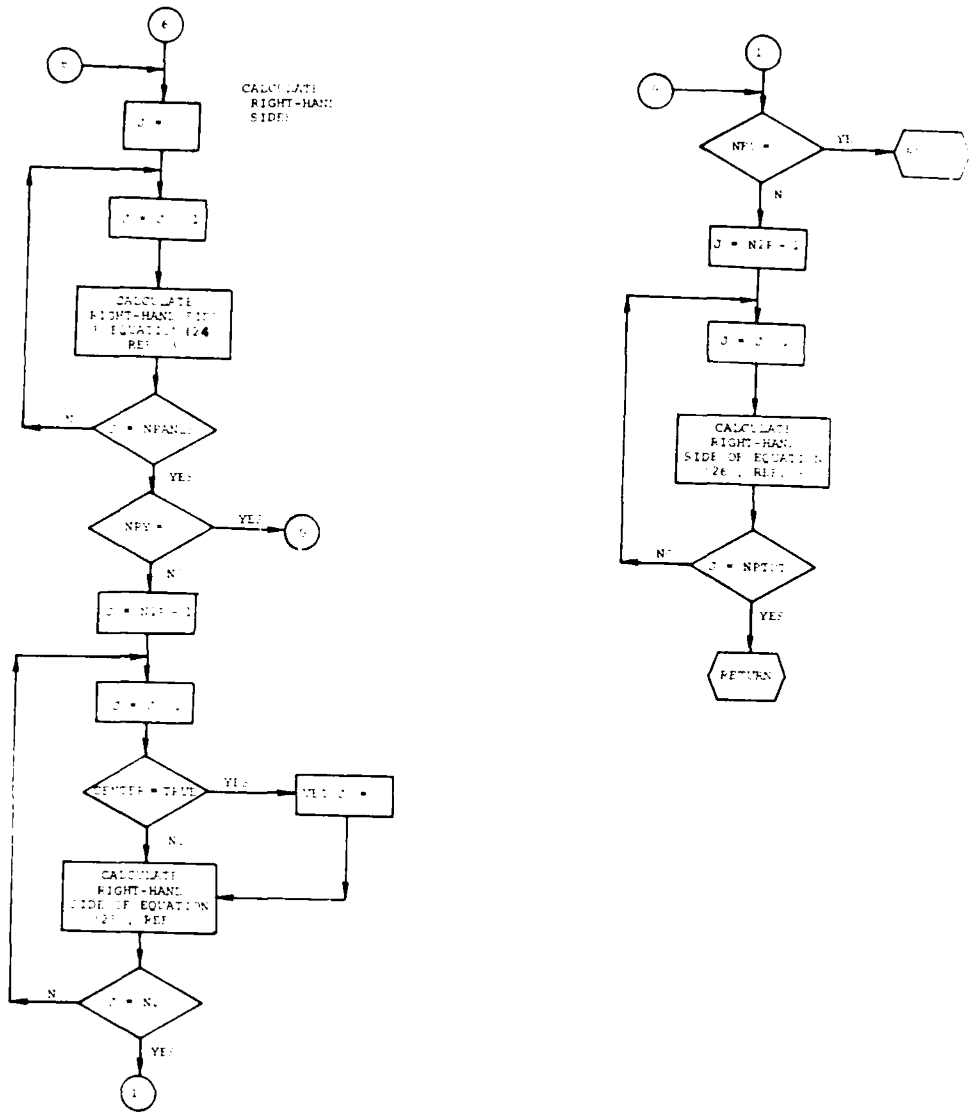


(a)
Figure A-6.- Flow chart of subroutine DPRHS.



(b)

Figure A-6.- Continued.



(c)

Figure A-6.- Concluded.

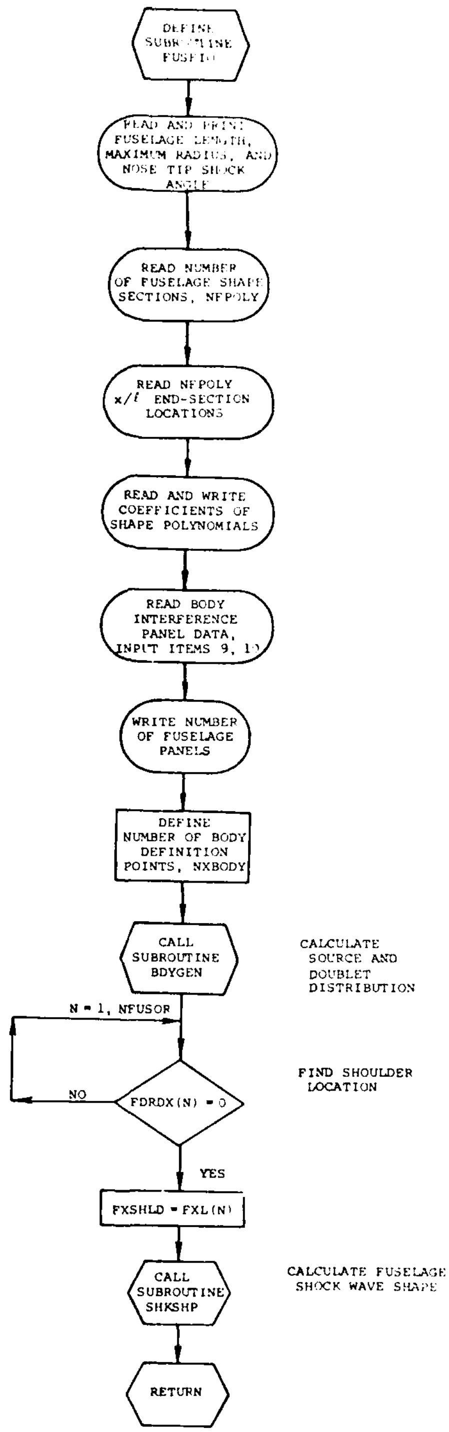
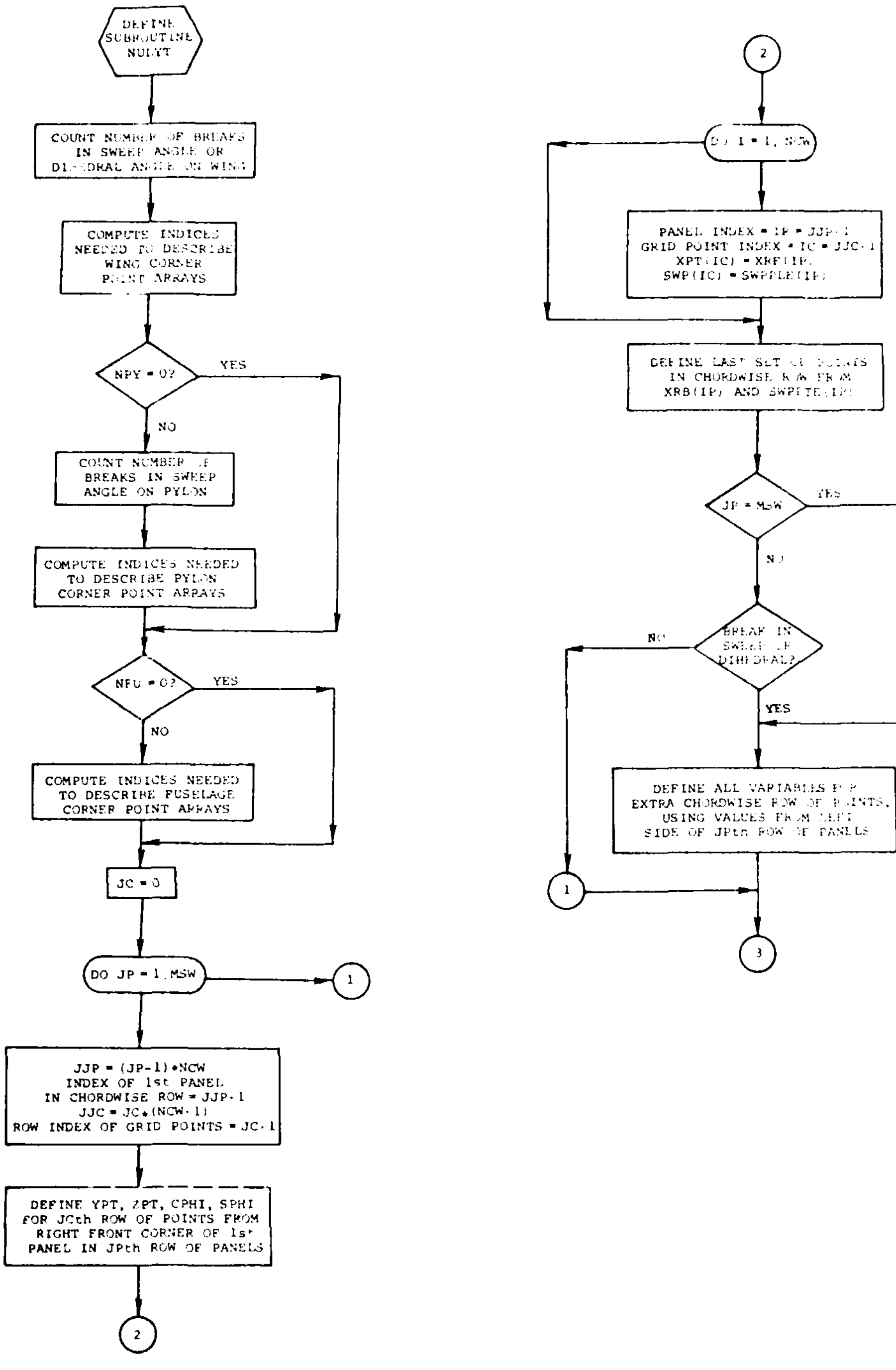
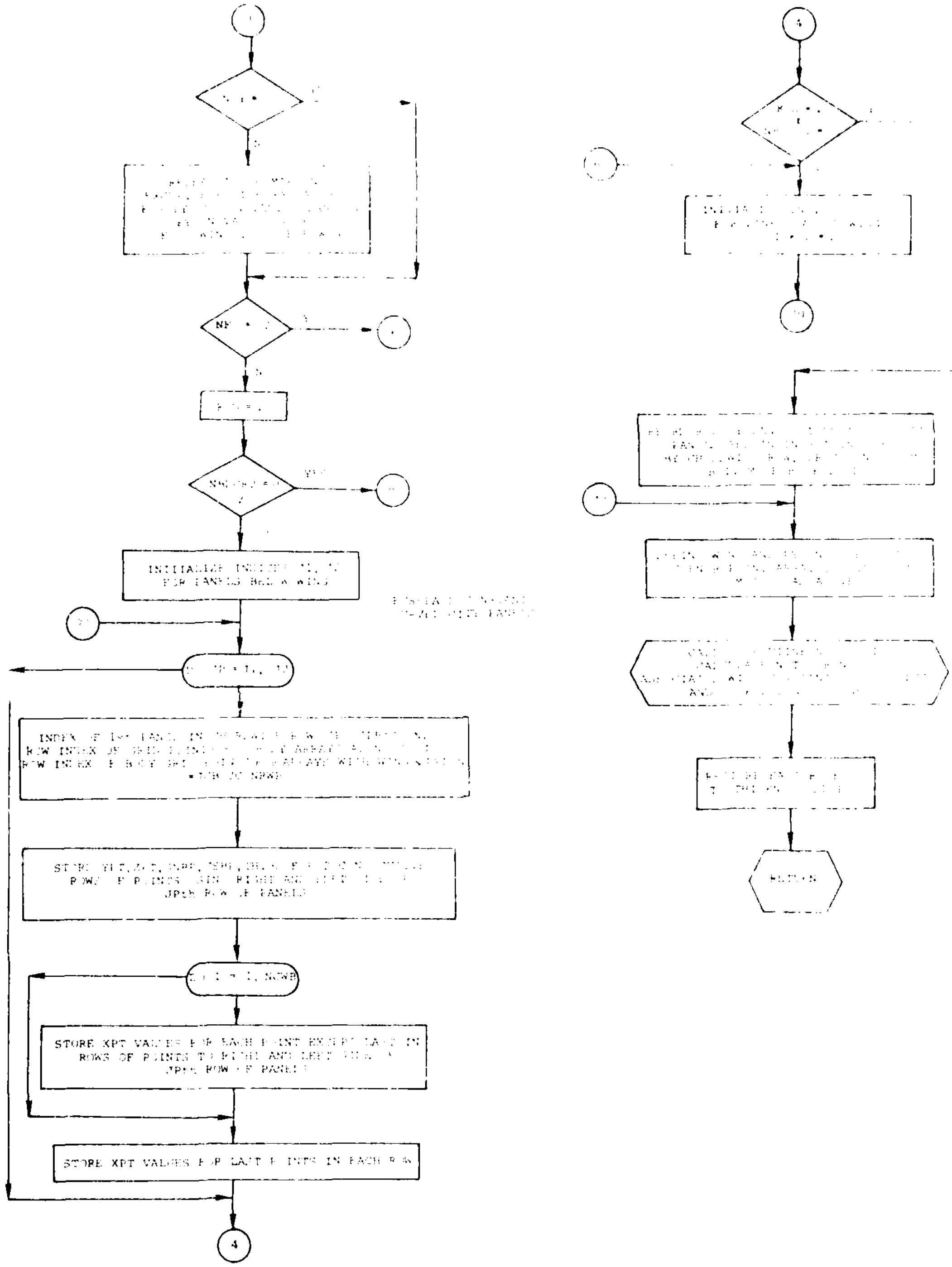


Figure A-7.- Flow chart of subroutine FUSEIO.



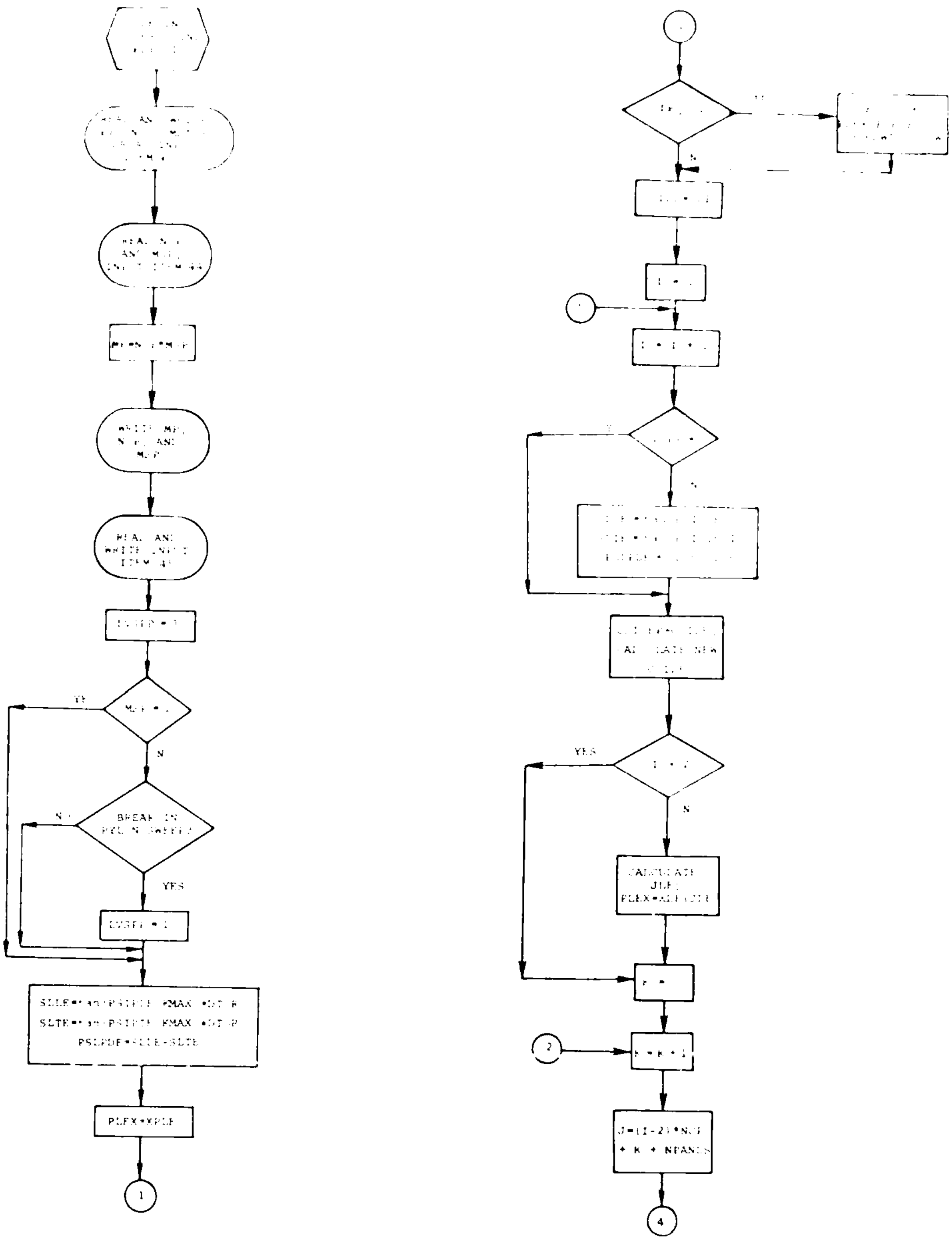
(a)

Figure A-8.- Flow chart of subroutine NULYT.

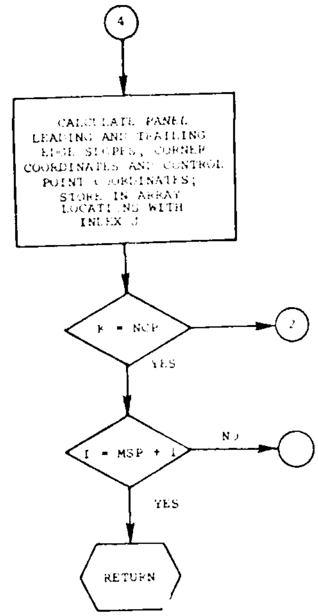


(b)

Figure A-8.- Concluded.



(a)
Figure A-9.- Flow chart of subroutine PLYOUT.



(b)

Figure A-9.- Concluded.

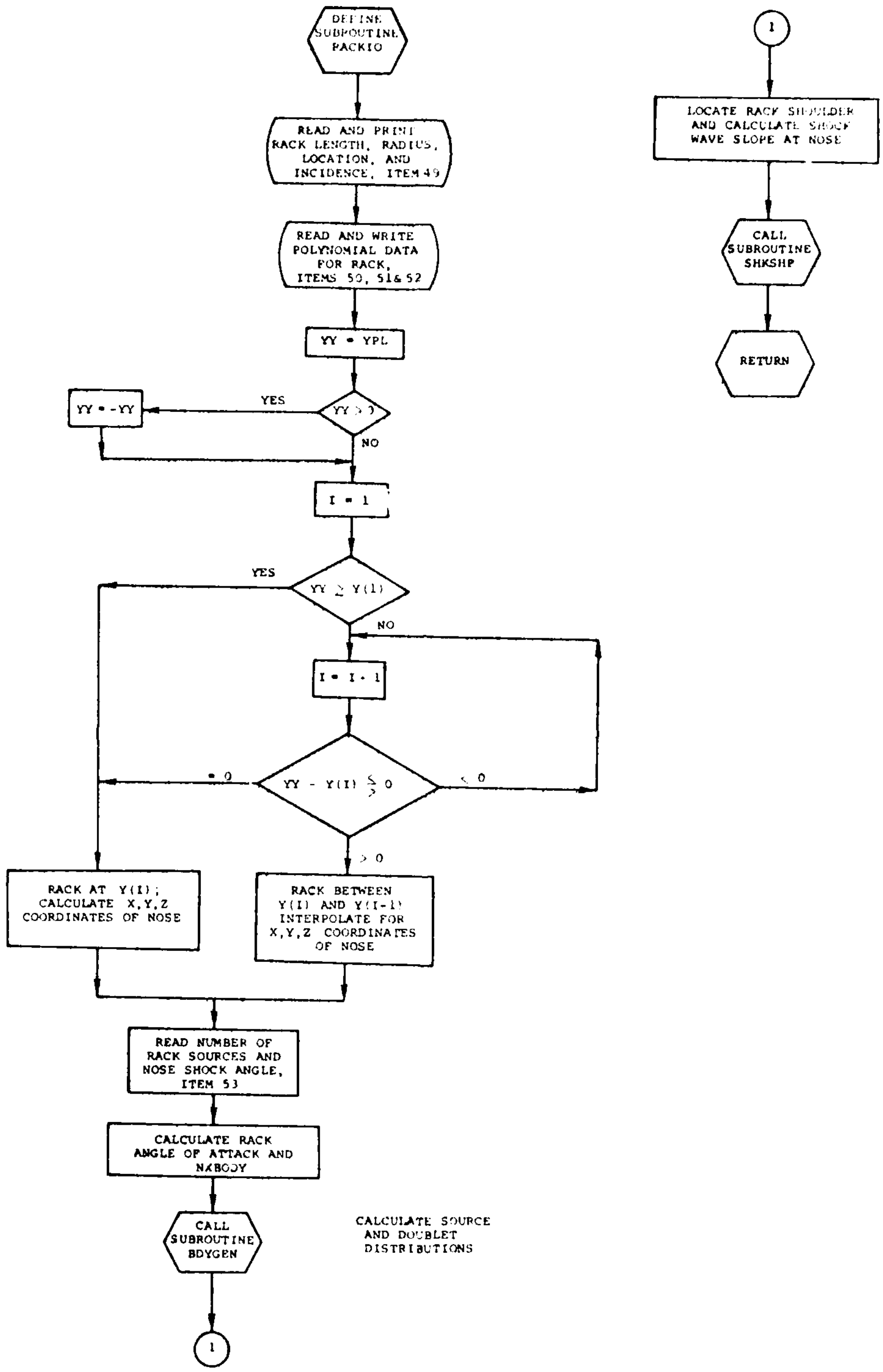


Figure A-10.- Flow chart of subroutine RACKIO.

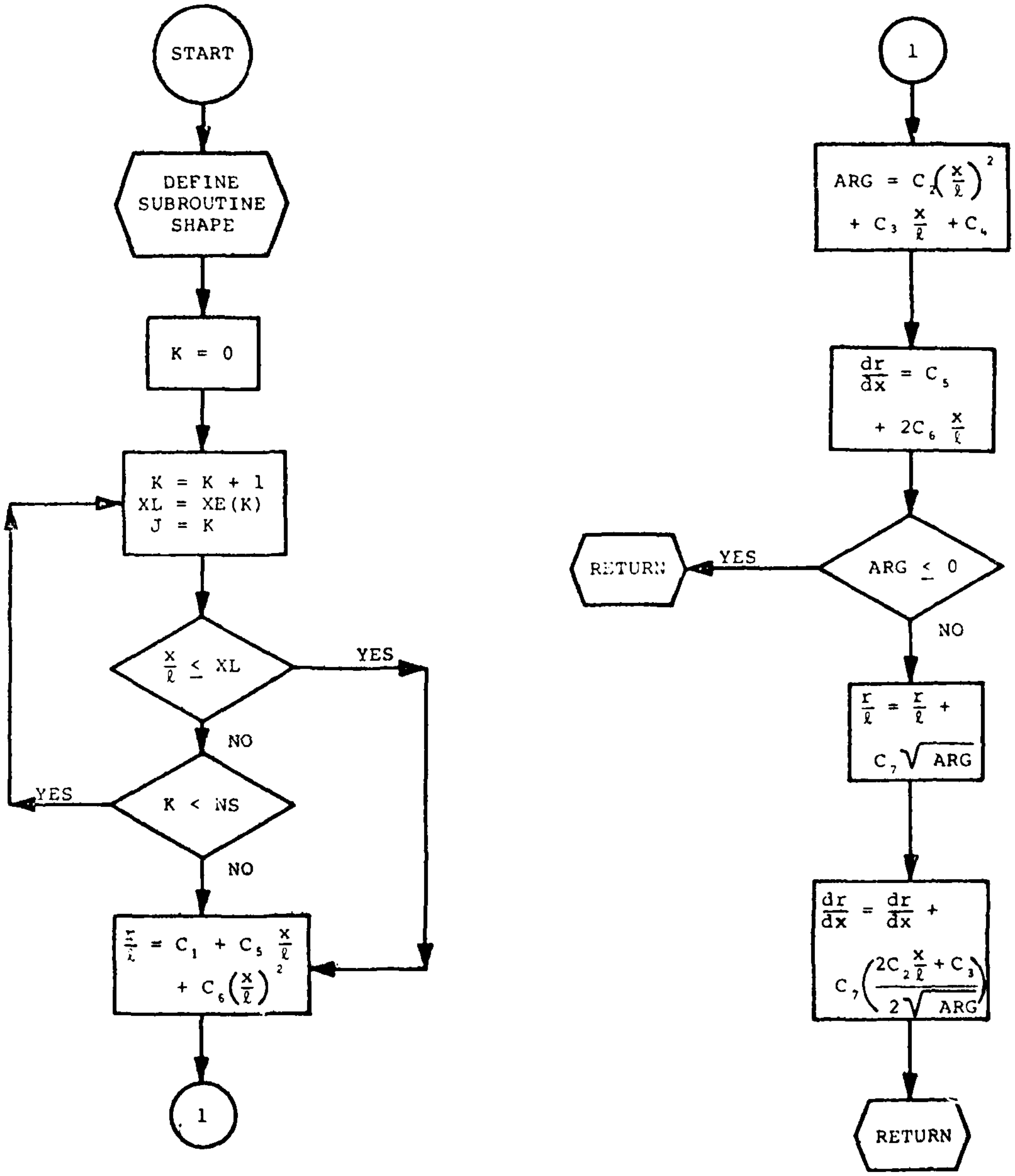
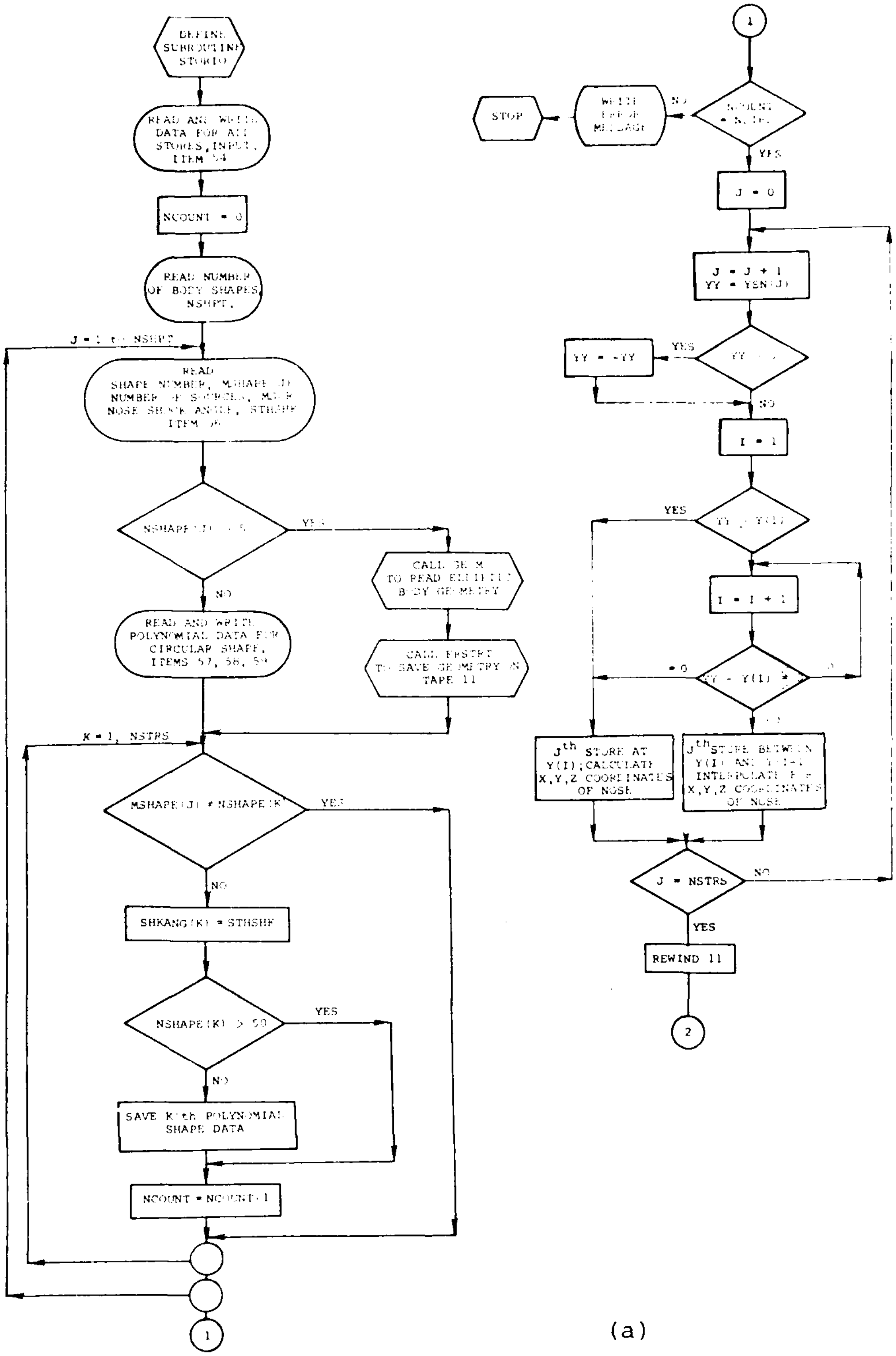
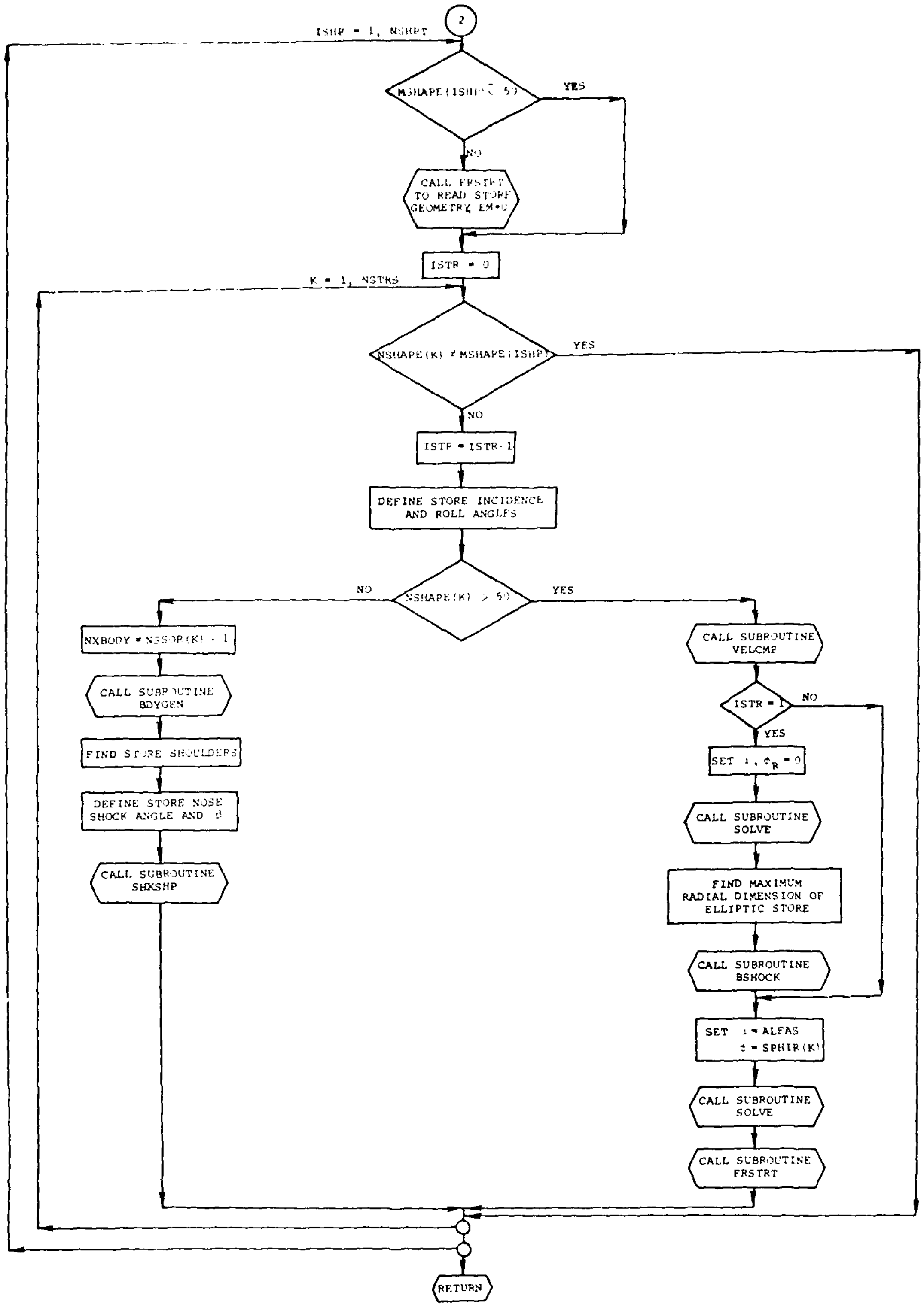


Figure A-11.- Flow chart of subroutine SHAPE.

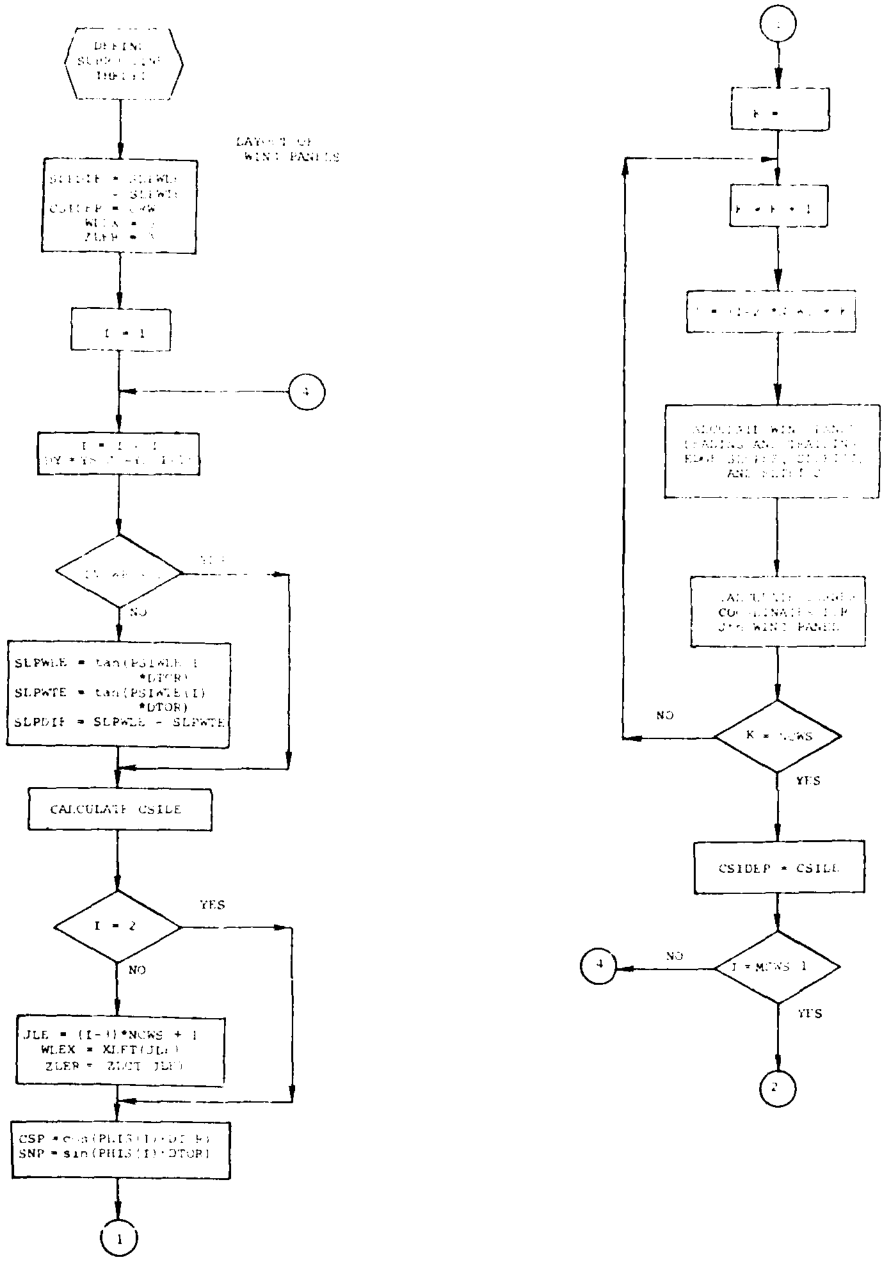


(a)

Figure A-12.- Flow chart of subroutine STORIO.

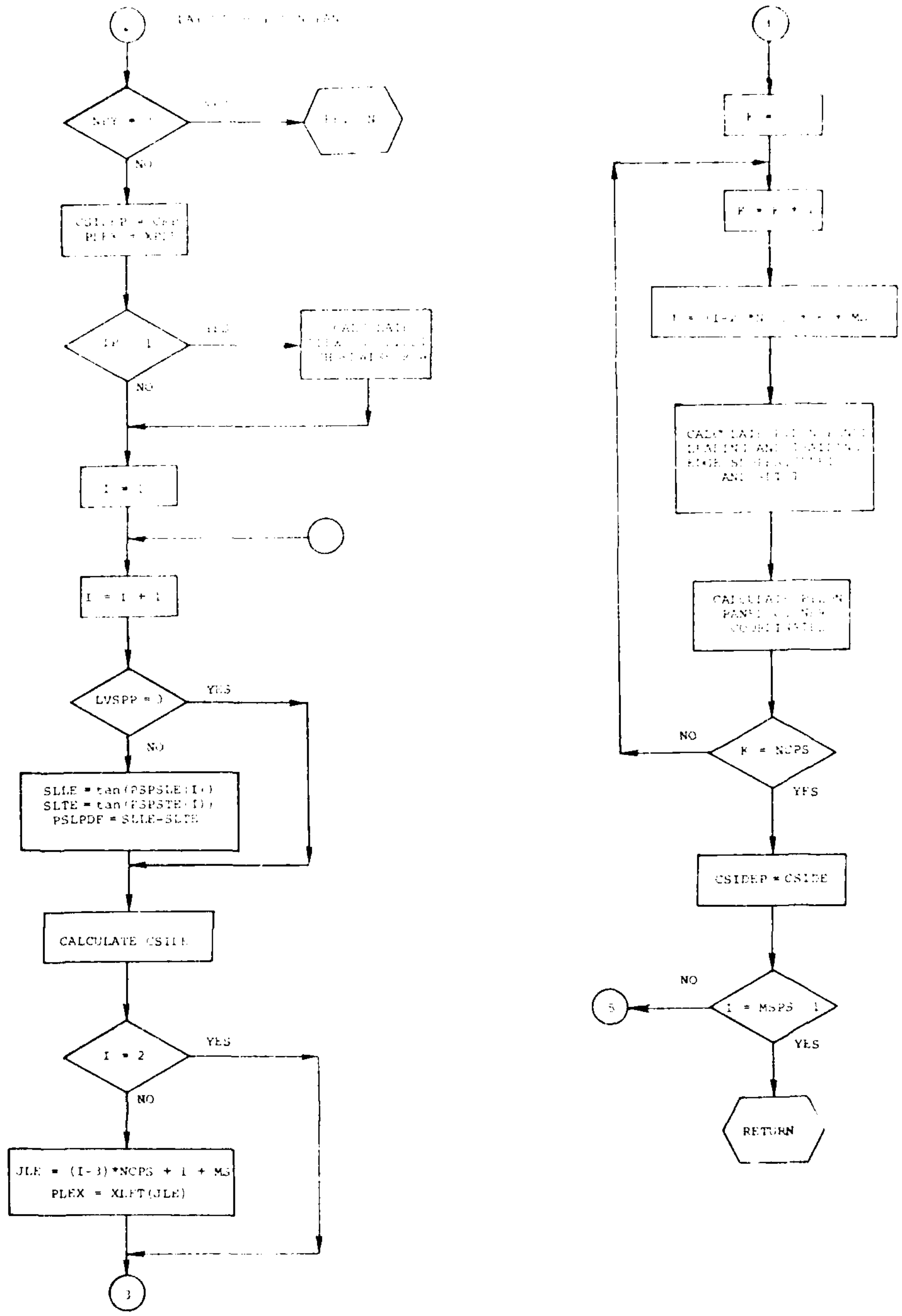


(b)
Figure A-12.- Concluded.



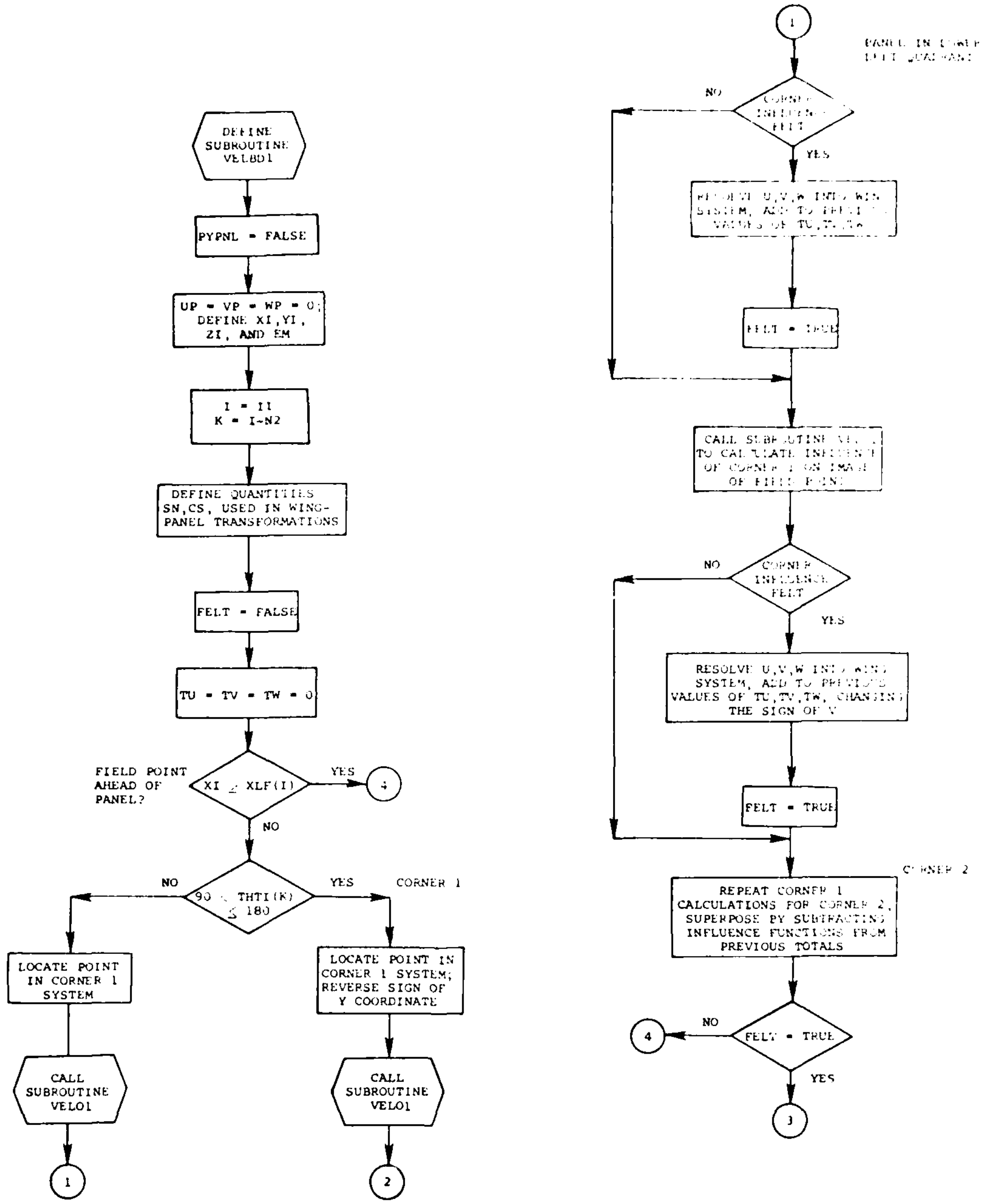
(a)

Figure A-13.- Flow chart of subroutine THKLYT.



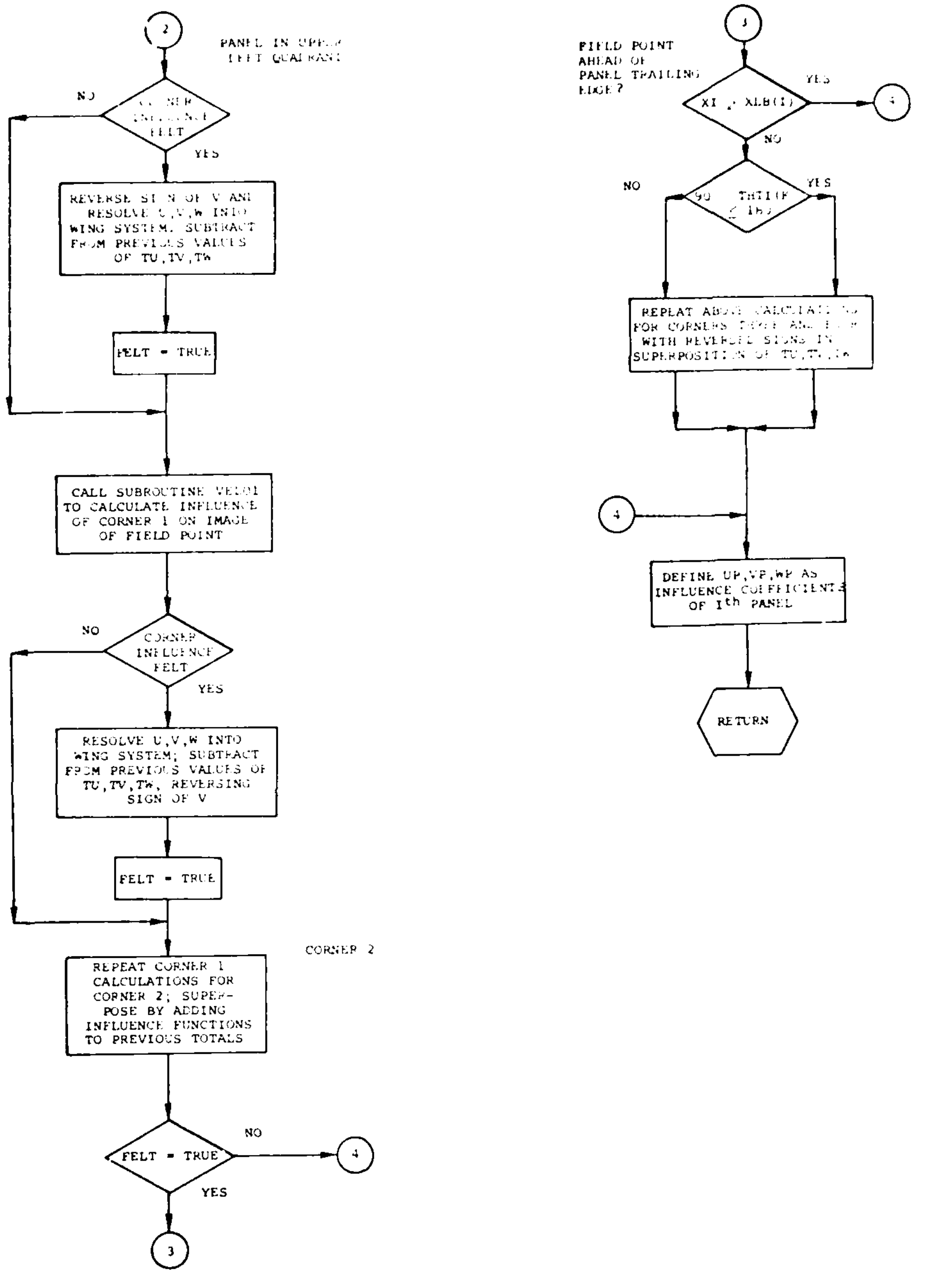
(b)

Figure A-13.- Concluded.



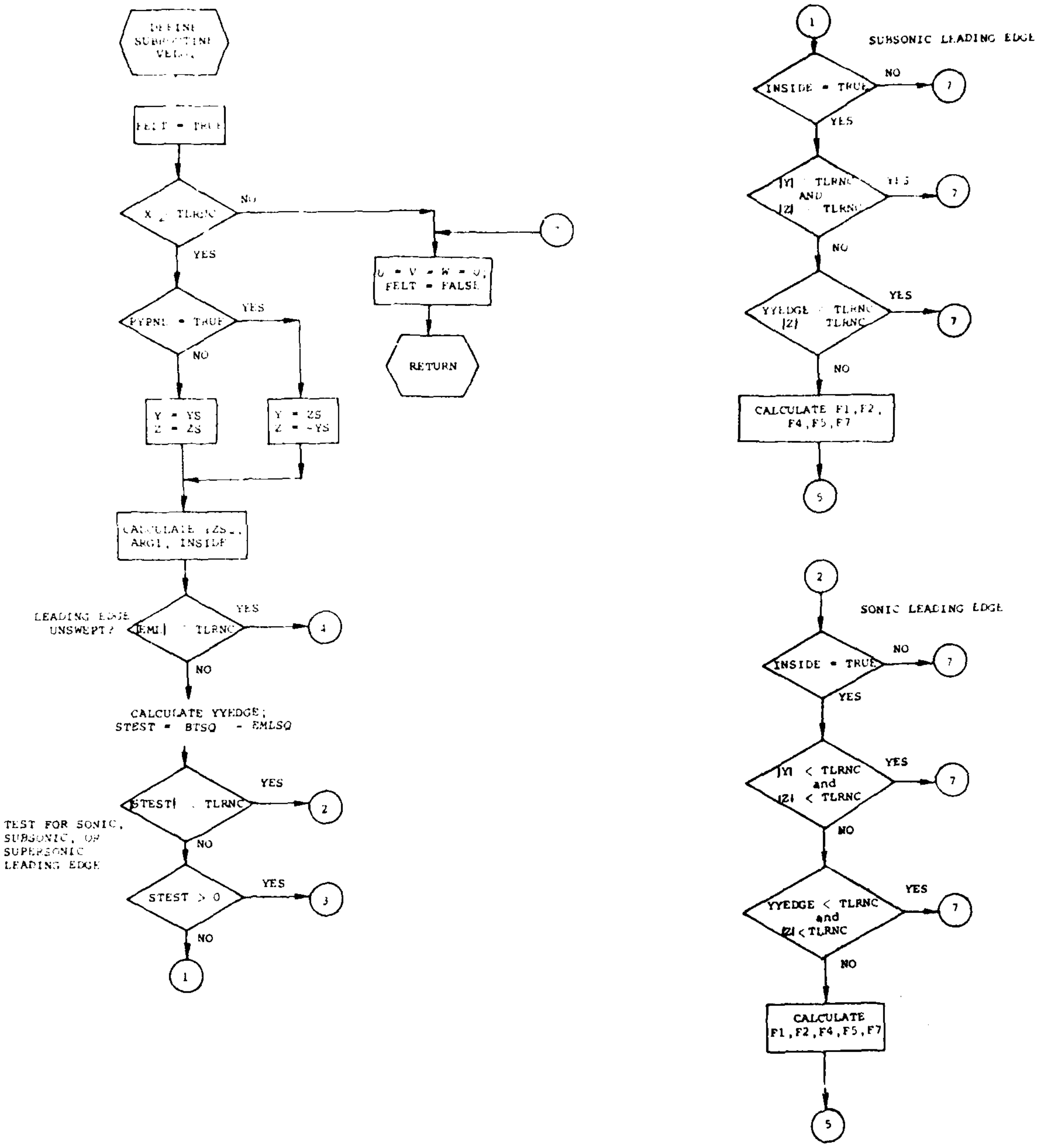
(a)

Figure A-14.- Flow chart of subroutine VELBD1.



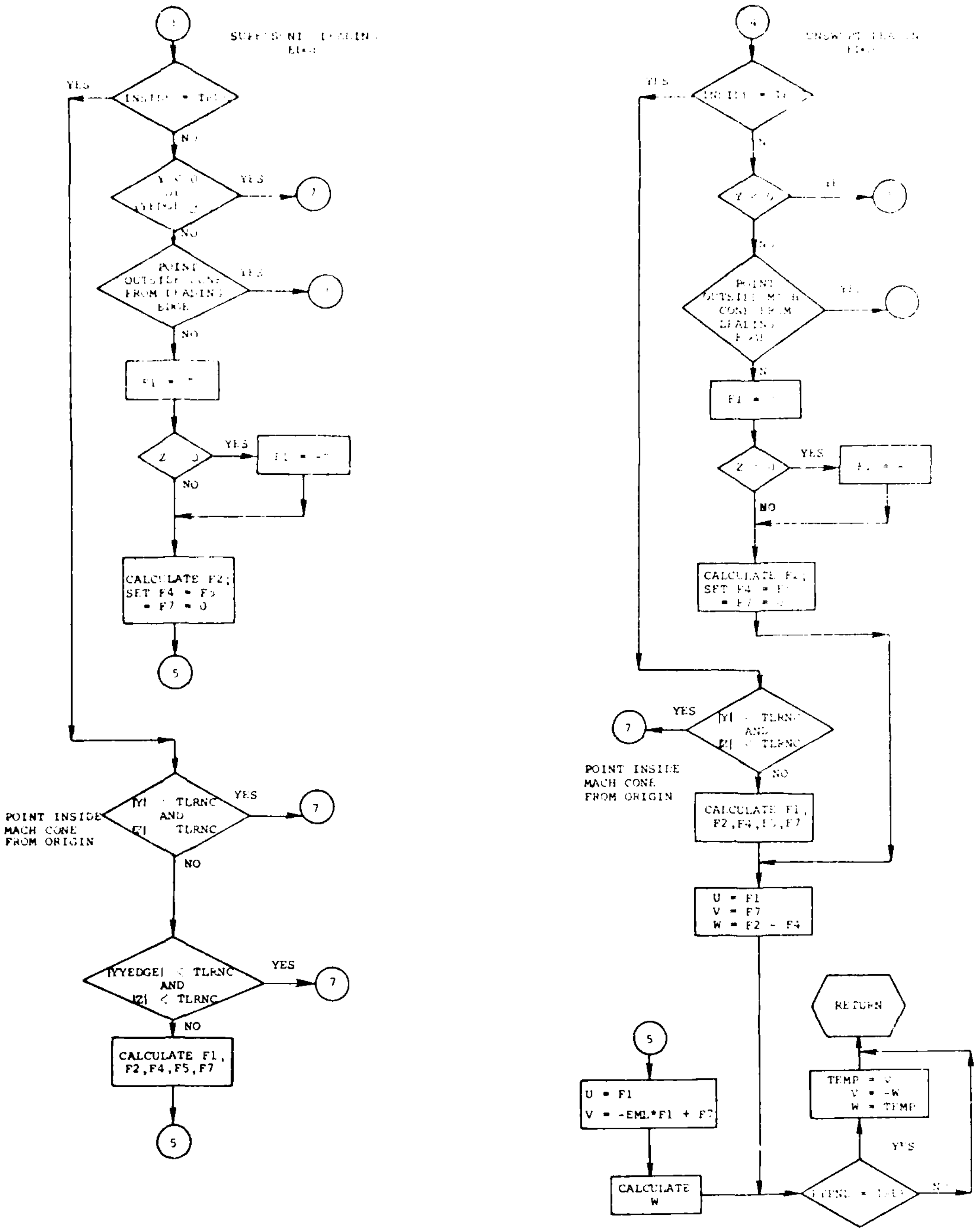
(b)

Figure A-14.- Concluded.



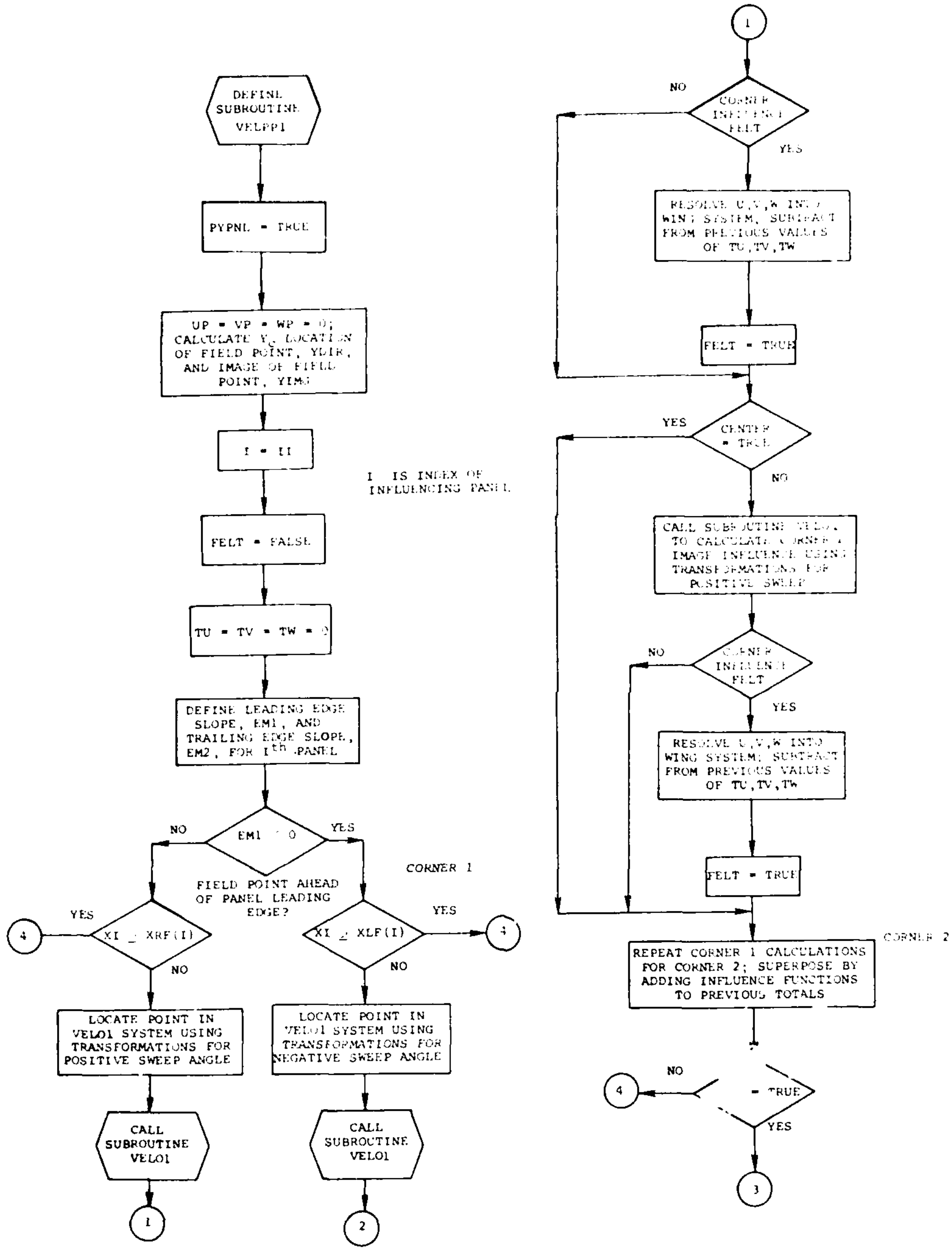
(a)

Figure A-15.- Flow chart of subroutine VELO1.



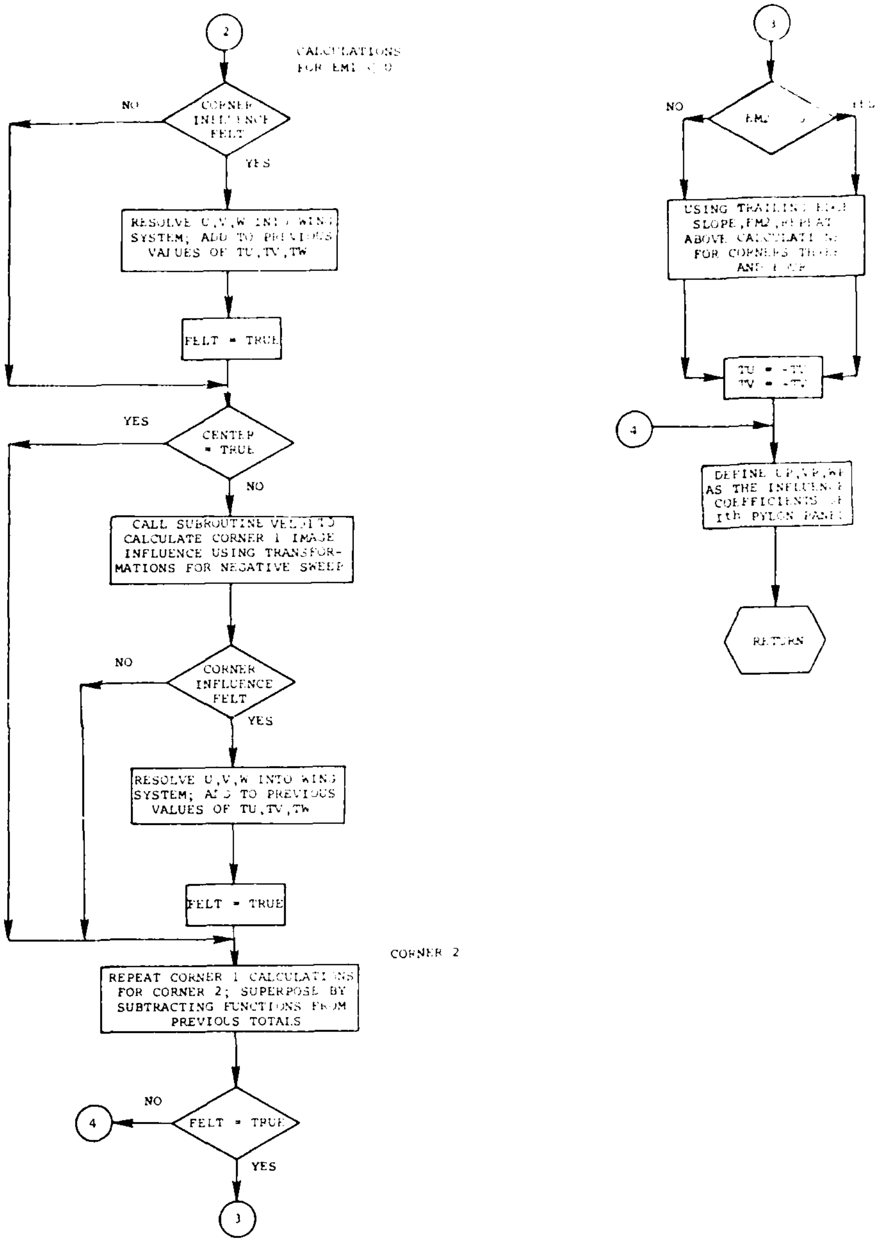
(b)

Figure A-15.- Concluded.



(a)

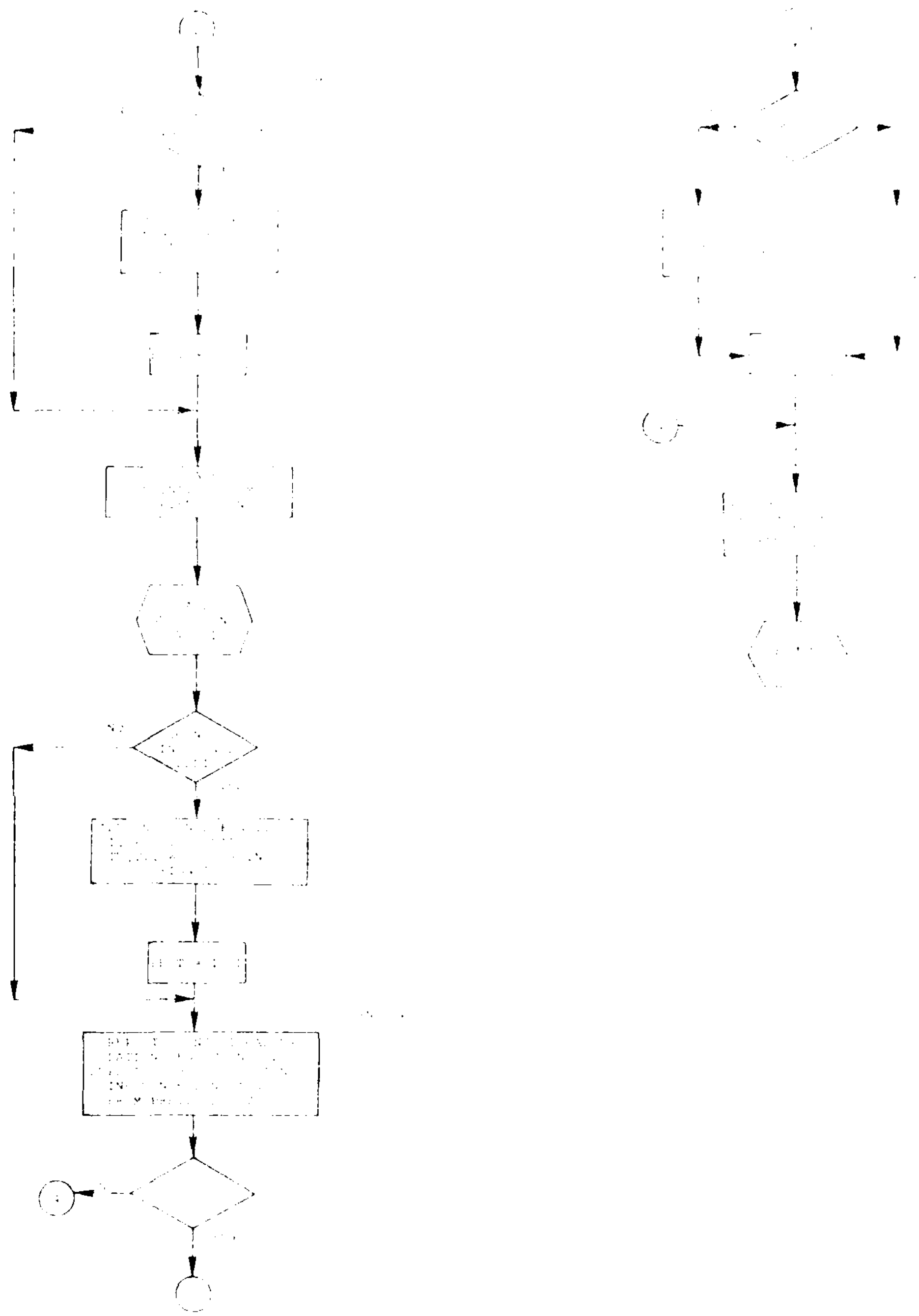
Figure A-16.- Flow chart of subroutine VELPP1.



(b)

Figure A-16.- Concluded.

(a)
Figure A-17.- Flow chart of subroutine VELWP1.



(b)
Figure A-17.- Concluded.

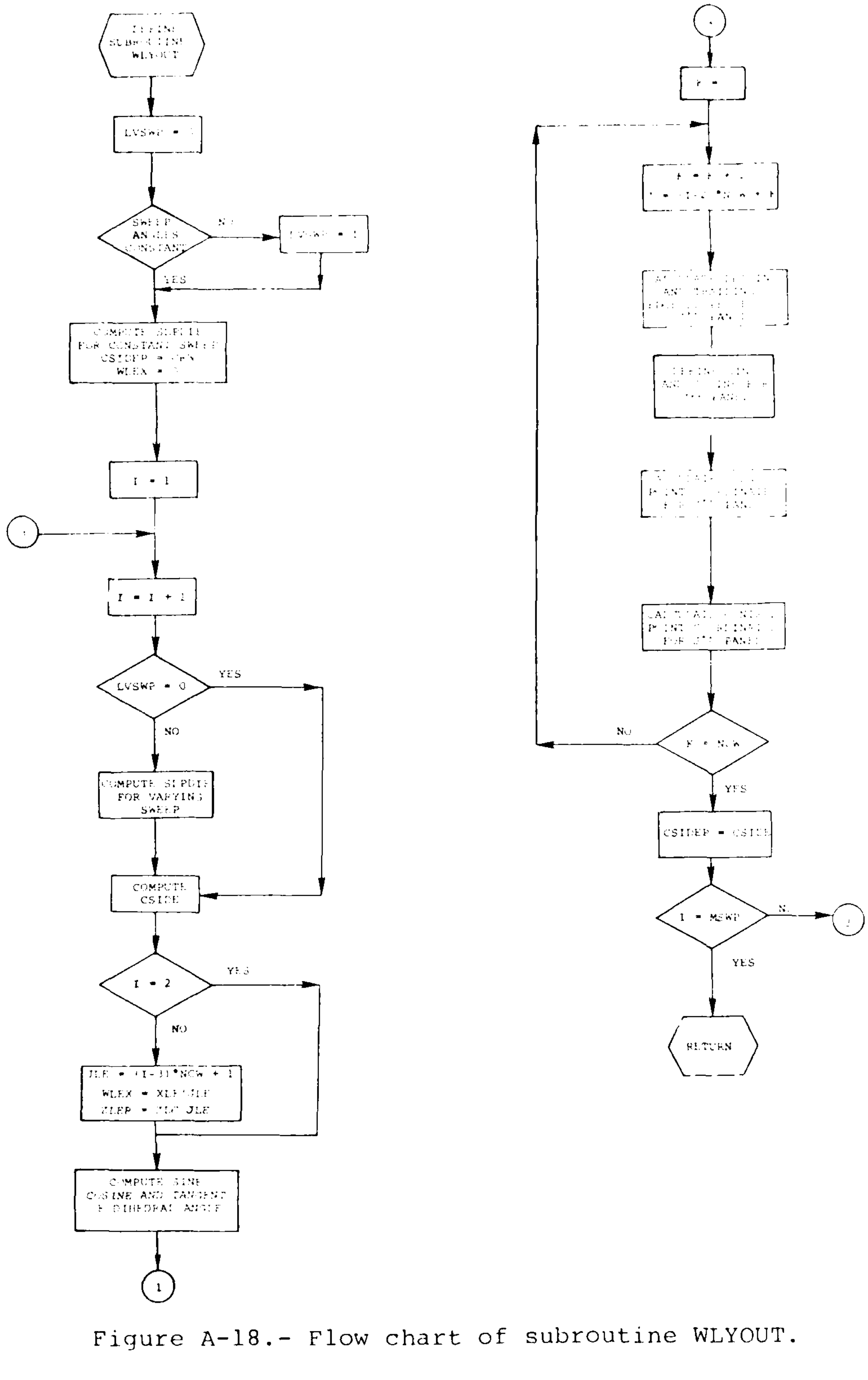


Figure A-18.- Flow chart of subroutine WLYOUT.

APPENDIX B

COMMON BLOCK DESCRIPTIONS - PROGRAM I

B-1 Introduction

The purpose of this Appendix is to provide more detailed information on the variables passed between routines through common blocks in Program I. This appendix will present the tables equating program notation to the algebraic notation and variable descriptions. The common blocks are arranged alphabetically. It is followed by a section detailing the special usage of blank common. When a description identifies an item as an input, consult Volume II of this report for further definitions. For variables used both in the description of the noncircular fuselage and elliptic stores, only the input item for the fuselage will be referenced.

A cross-reference chart showing the routines and the common statements contained in each is presented in Figure B-1. Across the top of the chart are the subroutine names including the main program LDCALC. Down the side of the page are the common names. The last one is blank common.

B-2 Description of Variables in Labeled Commons

Var.	Engr. Symbol and Description

COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7), SSUMK(7),SSUMKD(7)

FBASE	fuselage x-station at which base singularities originate, feet
FSUMK	strength of source originating at fuselage base
FSUMKD	strength of doublet originating at fuselage base
RBASE	rack x-station at which base singularities originate, feet
RSUMK	strength of source originating at rack base
RSUMKD	strength of doublet originating at rack base

SBASE	store x-station at which base singularities originate, feet
SSUMK	strength of source originating at store base
SSUMKD	strength of doublet originating at store base

COMMON /BGEOM/ XFUS(51),ZFUS(51),FUSARD(51),FUSBY(51),FUSAZ(51),
XJ(51),PHIK(33)

XFUS(I)	x_B coordinate of Ith station used to define body external geometric shape, feet; input item 18
ZFUS(I)	z_B coordinate of Ith station containing cambered offset, feet; input item 19
FUSARD(I)	cross sectional area at Ith station, ft^2; input item 22
FUSBY(I)	b_y, elliptic horizontal semi-axis (y-direction), feet; input item 23
FUSAZ(I)	a_z, elliptic vertical semi-axis (z-direction), feet; input item 24
XJ(J)	x_B coordinate of Jth station used to define body panel corners, feet; input item 32
PHIK(J)	ϕ_k, polar angle defining Jth meridian of panel edges, degrees; input item 31

COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI,
XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)

NINLET	number of open inlet panels; input item 14
NINVEL	number of additional panels to be used in velocity calculations for inlet panels; input item 14
NTINL	total number of inlet panels to be used in a given calculation
RVIVO	inlet mass flow ratio; ratio of open inlet panel frontal area to total inlet panel area
NINBLK	number of blocked inlet panels; input item 14
BTINLT	β associated with inlet panels; input item 29

YCPI — y-location of inboard most edge of inlet panels

XINLT, YINLT, ZINLT — coordinates of outboard leading edge of inlet panels used to locate center of inlet shock propagation

ZINLTE, YINLTE, ZINLTE — coordinates of outboard trailing edge of inlet panels used to locate lower lip of inlet and turning point of shock

JINLT(I) — fuselage panel number associated with Jth inlet panel

COMMON /BINSHK/ NIS,XISHLD,EALPI,XCLOSD,MAXSHI,NINL(8),PHINL(8),
    YINL(8),XINL(80),RINL(80)

NIS — number of inlet shock traverses; input item 28

XISHLD — x-station at which [illegible]'s associated with inlet return to free stream

EALPI — angle of attack correction factor for inlet shock (EALPHA)

XCLOSD — x-location of leading edge of blocked inlet panels

MAXSHI — maximum number of points in inlet shock tables per traverse (=MAXSHK)

NINL(I) — number of points computed for Ith shock traverse

PHINL(I) — angle measured from z-axis below inlet of Ith inlet shock traverse, positive counterclockwise

YINL(I) — y-station of Ith inlet shock traverse

XINL — table of x-values of inlet shock

RINL — table of radial values of inlet shock

COMMON /BIPYZ/ XBIP,NYZBIP,YCB(33),ZCB(33)

XBIP — x-station at which the y,z geometry used in the definition of the body interference shell are obtained for the noncircular fuselage, feet

NYZBIP — number of y,z values used to define panel corners around the circumference of the interference shell; NYZBIP=KRAD of local body segment

YCB(I)	y-coordinates of noncircular interference shell, feet
ZCB(I)	z-coordinates of noncircular interference shell, feet

COMMON /BLKPAN/ COST,SINT,XBTJ,YBTJ,ZBTJ,XC1,YC1,ZC1,XPTI,YPTI,
ZPTI,COSTI,SINTI,COSD,SIND,LZERO

COST,SINT	$\cos(\theta_j)$ and $\sin(\theta_j)$, cosine and sine of polar angle of Jth influencing source panel
XBTJ	XPT_j coordinate of Jth influencing control point, feet
YBTJ	YPT_j coordinate of Jth influencing control point, feet
ZBTJ	ZPT_j coordinate of Jth influencing control point, feet
XC1	x_B coordinate of Jth panel reference corner, feet
YC1	y_B coordinate of Jth panel reference corner, feet
ZC1	z_B coordinate of Jth panel reference corner, feet
XPTI	XFT_i coordinate of Ith influenced field point, feet
YPTI	YFT_i coordinate of Ith influenced field point, feet
ZPTI	ZFT_i coordinate of Ith influenced field point, feet
COSTI,SINTI	$\cos(\theta_i)$ and $\sin(\theta_i)$, cosine and sine of polar angle at Ith influenced panel or field point
COSD,SIND	$\cos(\delta_i)$ and $\sin(\delta_i)$, cosine and sine of panel incidence angle at Ith influenced panel or field point
LZERO	PANVEL angle calculation option: F=yes, T=no

COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4),XI,YI,XJ,
ZJ,BETA0,BETAL,SUBSON,SUPERS

AMACH	Mach number used in source panel influence calculation

TAND	$\tan\delta_j$, tangent of incidence angle of Jth influencing panel
CX	panel chord length, feet
XCOR(K)	x of Kth corner in local panel system, feet
YCOR(K)	y of Kth corner in local panel system, feet
ZCOR(K)	z of Kth corner in local panel system, feet
XI	x of Ith field point in local panel system, feet
YI	y of Ith field point in local panel system, feet
ZI	z of Ith field point in local panel system, feet
XJ	x of Jth panel control point in panel system, feet
ZJ	z of Jth panel control point in panel system, feet
BETA0	$\sqrt{M_\infty^2-1}$, free stream Mach number constant
BETAL	$\sqrt{M_\ell^2-1}$, local Mach number constant
SUBSON	subsonic logical indicator; SUBSON = AMACH.LT.1
SUPERS	supersonic logical indicator; SUPERS = AMACH.GT.1

```
COMMON /BOPTNS/ J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV,
     INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE,REFA,REFD,REFL,REFX,
     CCTEST,ITMAX,BODL,IZ1(12)
```

J0	reference area indicator; see input item 16
J2	body type indicator; see input item 16
J6	body camber indicator; see input item 16
NFUS	number of body segments; see input item 16
NRADX(I)	number of points used to define section of Ith body segment
NFORX(I)	number of axial station on Ith body segment
J2TEST	parameter to specify body camber and cross-section definition
IPRES	not used
ISOLVE	not used

INLET	logical inlet indicator: true = inlet panels present false = no inlet panel present
IPLOT	not used
IPRT	optional print control parameter, see input item 14
IUVW	component velocity calculation option; see input item 14
XSTART	x-station at which pressure integration is started, feet
XWLE	x-station at which pressure integration is ended, feed
REFA	body reference area, ft^2; see input item 17
REFD	body reference length used for moment normalization, feet; see input item 30
REFL	body length, feet; see input item 30
REFX,REFZ	x,z coordinates of moment reference point, feet; see input item 30
CCTEST	solution convergence control criteria (=0.0001)
ITMAX	solution maximum number of iterations (=20)
IZ1	dummy array, not used

```
COMMON /BPGEOM/ SNT2(200),CST2(200),ZLC(200),ZRC(200),THTI(200),
     SANGW
```

SNT2(J)	$\sin(ANG_J)$ associated with Jth body panel
CST2(J)	$\cos(ANG_J)$ associated with Jth body panel
ZLC(I)	z_W coordinate of left edge of Ith constant u-velocity panel
ZRC(I)	z_W coordinate of right edge of Ith constant u-velocity panel
THTI(J)	$(\pi - THT_J)$;THT_J is value of THT associated with Jth panel
SANGW	sin(ANGW) = ZBWO/FRMAX

```
COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,DBETA,
     EALPHA,CNU0,CNU2,XSHLDR,SHK(3),XSHK(100),RSHK(100)
```

NSHK(I)	number of points used to represent Ith modified shock shape
PHIS(I)	polar angle at which Ith modified shock is computed, degrees; input item 33
THETN(I)	nose limited shock angle of initial shock shape at Ith polar angle, degrees
MAXSHK	maximum number of points in Ith shock shape; input item 14
NSHOCK	number of modified shock shape computed; input item 14
DBETA	not used
EALPHA	angle of attack correction to shock shape; input item 15
CNU0,CNU2	not used
XSHLDR	x_B location of body nose shoulder, feet; input item 15
SHK	dummy array, not used
XSHK,RSHK	arrays containing x_B and r_B locations of NSHOCK sets of NSHK(I) points representing the modified nose shock shape, feet

```
COMMON /CAMBER/ ALPHAL(200)
```

ALPHAL(J)	$\tan(\alpha_\ell)$, slope of wing camberline at the constant u-velocity panel control points; input item 39

```
COMMON /CONFIG/ NFU,NPY,NSTRS,LVSWP,NRACK
```

NFU	fuselage indicator; input item 4
NPY	pylon indicator; input item 4
NSTRS	number of stores indicator; input item 4

LVSWP	breaks in wing sweep indicator; LVSWP=0, no; LVSWP=1, yes
NRACK	rack indicator; input item 4

```
COMMON /CONSTS/ PI,PI2,DTOR,RTOD,FOURPI
```

PI	π
PI2	$\pi/2$
DTOR	$\pi/180°$
RTOD	$180°/\pi$
FOURPI	4π

```
COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT,
     NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5),IXZSYM,
     NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7,IAR,IAN,IUB,
     IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB,NAG,NAP,NAV,NAS,
     NASHK,NAFLD,IA0,ID0,ISK0,IYIM,IZIM,ISVN,ISKP,NRING,IROW(50)
```

NX	sum of axial geometry stations (= $\sum^{NFUS}$ NFORX(I))
NR	not used
KX	sum of axial panel geometry stations (= $\sum^{NFUS}$ KFORX(I))
KR	not used
NXNR	sum (= $\sum^{NFUS}$ NFORX(I)*NRADX(I))
KXKR	sum (= $\sum^{NFUS}$ KFORX(I)*KRADX(I))
NXKR	sum (= $\sum^{NFUS}$ NFORX(I)*KRADX(I))
MAXNX	maximum of (NFORX(I),I=1,NFUS)
MAXKR	maximum of (KFORX(I),I=1,NFUS)
NATOT	last location accessed in blank common
NBODY	number of body panels
KFUS	number of body segments paneled (=NFUS)
KRADX(I)	number of meridian lines used to define panel edges on Ith body segment; input item 27

KFORX(I)	number of axial stations used to define leading and trailing edges of panels on Ith body segment; input item 27
IXC(I)	location in blank common of start of XC array for Ith body segment
IYC(I)	location in blank common of start of YC array for Ith body segment
IZC(I)	location in blank common of start of ZC array for Ith body segment
IXZSYM	XZ-plane symmetry option; input item 14
NADIM	dimensioned length of A array in blank common
NXDIM	maximum allowable number of axial stations (=51)
NG	last location in blank common containing source panel geometry arrays
IXPT	location in blank common of start of control points, XPT
IYPT	location in blank common of start of control points, YPT
IZPT	location in blank common of start of control points, ZPT
ITH	location in blank common of start of array THET
IDEL	location in blank common of start of array DELTA
NTAP7	number of variables last written on TAPE7
IAR	location in blank common of start of panel areas, AREA
IAN	location in blank common of start of temporary array AN containing influence coefficients
IUB	location in blank common of start of temporary array UB containing U,V,W influence coefficients
IGB(IALP)	location in blank common of start of IALPth array containing the source strength solution, GB
IVB	location in blank common of start of temporary array, VB, containing normal velocity boundary conditions

IU	location in blank common of start of U velocities
IV	location in blank common of start of V velocities
IW	location in blank common of start of W velocities
IVA,IWA	location in blank common of start of additional temporary velocity arrays
ICP	location in blank common of start of pressure coefficient, CP, array
IPHI	location in blank common of temporary PHI array
IYB	location in blank common of start of coordinates YB and ZB of temporary cross section geometry in NEWRAD
NAG	maximum locations in blank common required in GEOM
NAP	not used
NAV	maximum locations in blank common required in VELCMP
NAS	maximum locations in blank common required in SOLVE
NASHK	maximum locations in blank common required in BSHOCK
NAFLD	maximum locations in blank common required in FLDVEL
IA0	offset location in blank common of all above arrays when multiple configurations are simultaneously in core
ID0	offset location in blank common of ID array containing /DIMENS/ information for additional source panel configurations in core
ISK0	offset location in blank common of array containing information in common /BSHOCK/ for additional source panel configurations in core
IYIM,IZIM	location in blank common of start of image store Y and Z control point coordinates
ISVN,ISKP	location in blank common of start of temporary arrays, SVN and LSKP
NRING	number of rings of panels on body
IROW(I)	number of panels around Ith ring of panels

COMMON /EXVEL/ UEI(200),VEI(200),WEI(200),CIR(200)

UEI(I)	$u_{W_{i,\nu}}/V_\infty$ at the ν = Ith control point
VEI(I)	$v_{W_{i,\nu}}/V_\infty$ at the ν = Ith control point
WEI(I)	$w_{W_{i,\nu}}/V_\infty$ at the ν = Ith control point
CIR(I)	array containing right-hand sides of u-velocity equations

COMMON /FLOW/ ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ,FMCHSQ

ALFACR	α_f, radians
GAMF	not used
FMACH	M_∞; input item 3
RHO	not used
VINF	not used
BETA	$\beta = \sqrt{M_\infty^2-1}$
BETASQ	β^2
FMCHSQ	M_∞^2

COMMON /FSGEOM/ FRMAX,SSPAN,FLTHC,BODYPL

FRMAX	fuselage maximum radius; input item 5
SSPAN	wing semispan, feet; input item 35
FLTHC	fuselage length, feet; input item 5
BODYPL	body interference shell length, feet; input item 10

COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101)

NFSHK	number of x and r values in fuselage nose shock table generated from line singularities
FXSHK(I), FRSHK(I)	x and r coordinates of Ith circular fuselage nose shock location at zero angle of attack
FDRDX(J)	DR/DX at Jth circular fuselage control point

COMMON /FSOR/ FXL(101),FSS(100),FDS(100),NFSOR

FXL — array containing the x positions of the fuselage sources; positive, measured from tip of nose

FSS — array containing the strengths of the circular fuselage source distribution

FDS — array containing the strengths of the circular fuselage doublet distribution

NFSOR — number of circular fuselage sources and doublets; input item 9

COMMON /HEAD/ TITLE1(20),TITLE2(20)

TITLE1 — array containing hollerith description of non-circular body external geometry

TITLE2 — array containing hollerith description of non-circular body paneling distribution

COMMON /ICVEL/ UP,VP,WP,II,IF,DELTP(200)

UP — u/V_∞ perturbation velocity at a control point due to pylon or wing thickness; also u-velocity panel u influence function at a control point

VP — v/V_∞ perturbation velocity at a control point due to pylon or wing thickness; also u-velocity panel v influence function at a control point

WP — w/V_∞ perturbation velocity at a control point due to pylon or wing thickness; also u-velocity panel w influence function at a control point

II,IF — initial and final values, respectively, of thickness panel index

DELTP — $1/\pi(u_+/V_\infty)$; u-velocity panel strength

COMMON /INDEX/ NCW,MSW,MSWP,NPANLS,NCWB,NBDCR1,NBDCR2,NBD,NBIP, MP,NCP,MSP,N1P,N2,N2P,NPTOT

NCW — number of wing panels chordwise, input item 36

MSW — number of wing panels spanwise, input item 36

MSWP — MSW+1

NPANLS — number of u-velocity panels on left wing panel: NCW*MSW

NCWB	number of BIP rings, input item 9
NBDCR1	number of panels in ring above wing, input item 9
NBDCR2	number of panels in ring below wing, input item 9
NBD	total number of fuselage u-velocity panels in a ring on the half body
NBIP	number of u-velocity panels on fuselage; NBIP = NBD*NCWB
MP	number of u-velocity panels on pylon; MP = NCP*MSP
NCP	number of pylon panels chordwise, input item 44
MSP	number of pylon panels spanwise, input item 44
N1P	NPANLS + 1
N2	NPANLS + MP
N2P	N2 + 1
NPTOT	NPANLS + MP + NBIP

```
COMMON /NUINDX/ KW,KP,NRW,NRP,NRB,NPTW,NPTP,NPTB,NRTOT,NPTT,
     NRWS,NRPS,NPTWS,NPTPS,NRTOTS,NPTTS,NRWP,NPTWP,
     NCW1,NCP1,NCWB1,NCWS1,NCPS1
```

KW	number of breaks in sweep and/or dihedral on wing
KP	number of breaks in sweep on pylon
NRW	NRW = MSW + KW + 1
NRP	NRP = MSP + KP + 1
NRB	NRB = 2 * NBD
NPTW	NPTW = NCW + NRW + 1
NPTP	NPTP = NCP + NRW + 1
NPTB	NPTB = NCWB + NRB + 1
NRTOT	NRTOT = NRW + NRP + NRB
NPTT	NPTT = NPTW + NPTP + NPTB

NRWS	NRWS = NSWS + KW + 1
NRPS	NRPS = NSPS + KP + 1
NPTWS	NPTWS = NCWS + NRWS + 1
NPTPS	NPTPS = NCPS + NRPS + 1
NRTOTS	NRTOTS = NRWS + NRPS
NPTTS	NPTTS = NPTWS + NPTPS
NRWP	NRWP = NRW + NRP
NPTWP	NPTWP = NPTW + NPTP
NCW1	NCW1 = NCW + 1
NCP1	NCP1 = NCP + 1
NCWB1	NCWB1 = NCWB + 1
NCWS1	NCWS1 = NCWS + 1
NCPS1	NCPS1 = NCPS + 1

COMMON /PARAM/ XMACH,ALPHA,BETA,ALPHAC,PHIR,EM,SINAC,COSAC,
SPHI,CPHI,SINA,SINB

XMACH	Mach number seen by source panels
ALPHA	α, free stream angle of attack seen by source panels, degrees $(=\sin^{-1}(\sin(ALPHAC)\cdot\cos(PHIR))$
BETA	β, free stream angle of sideslip seen by source panels, degrees $(=\sin^{-1}(\sin(ALPHAC)\cdot\sin(PHIR))$
ALPHAC	α_c, included angle of attack seen by source panels, degrees
PHIR	ϕ_r, angle of roll seen by source panels
EM	temporary Mach number of last computation
SINAC	$\sin(\alpha_c)$
COSAC	$\cos(\alpha_c)$
SPHI	$\sin(\phi_r)$
CPHI	$\cos(\phi_r)$

AD-A099 331 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 20/4
PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLU--ETC(U)
NOV 80 J MULLEN, F. K GOODWIN, M F DILLENIUS F33615-76-C-3077
UNCLASSIFIED NEAR-TR-210-VOL-3 AFWAL-TR-80-3032-VOL-3 NL

3 OF 3
AD
A099331

END
DATE
FILMED
8-81
DTIC

SINA sin(α)

SINB sin(β)

COMMON /PYGEOM/ Z(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
SLLE,PSLPOF,CENTER,ZPL,LVSPP

Z(K)	z_p locations of pylon u-velocity panel side edges; input item 45
XPLE	location of pylon root chord leading edge; input item 43
YPL	y_w location of the pylon; YPL=Y(IP) of input item 37
CRP	pylon root chord; input item 43
HP	pylon height; input item 43
PSIPLE(K)	leading-edge sweeps of pylon u-velocity panels; input item 45
PHIPTE(K)	trailing-edge sweeps of pylon u-velocity panels; input item 45
IP	index of y_w location of the pylon; input item 43
SLLE	slope of pylon leading edge for Ith chordwise row of pylon panels
PSLPDF	difference in leading and trailing edge slopes PSLPDF = SLLE-SLTE for Ith chordwise row of pylon panels
CENTER	logical test for centerline pylon (CENTER=YPL.EQ.0)
ZPL	z_w location of the pylon; ZPL=ZLC((IP-2)·NCW+1)
LVSPP	breaks in pylon sweep indicator, LVSPP=0, no; LVSPP=1, yes

COMMON /RKGEOM/ RRMAX,RLTHC,XWROC,YWRO,ZERO,NRPOLY,RXEND(7),
RCOEF(7,7),XBRO,YBRO,ZBRO,RIBCR,SRIBCR,CRIBCR

RRMAX	maximum rack radius; input item 49
RLTHC	length of rack; input item 49

XWROC	x_w location of rack in wing coordinates, feet
YWRO	y_w location of rack in wing coordinates, feet
ZWRO	z_w location of rack in wing coordinates, feet
NRPOLY	number of rack shape polynomials; input item 50
RXEND(J)	rack polynomial endpoints; input item 51
RCOEF(I,J)	rack polynomial coefficients; input item 52
XBRO	x_B location of rack in body coordinates, feet
YBRO	y_B location of rack in body coordinates, feet
ZBRO	z_B location of rack in body coordinates, feet
RIBCR	RIC•DTOR; input item 49 for RIC
SRIBCR	sin(RIBCR)
CRIBCR	cos(RIBCR)

COMMON /RSHOCK/ NRSHK,RXSHK(50),RRSHK(50),RDRDX(101)

NRSHK	number of x and r values in rack nose shock table generated from line singularities
RXSHK(I), RRSHK(I)	x and r coordinates of Ith rack nose shock location at zero angle of attack, feet
RDRDX(J)	DR/DX at Jth rack control point

COMMON /RSOR/ RXL(101),RSS(101),RDS(101),NRSOR

RXL	array containing the x positions of the rack sources; positive, measured from tip of rack nose
RSS	array containing the strengths of the rack source distribution
RDS	array containing the strengths of the rack doublet distribution
NRSOR	number of rack sources and doublets; input item 53

COMMON /SHPDAT/ NFPOLY,FXEND(7),FCOEF(7,7)

NFPOLY	number of fuselage polynomials; input item 6
FXEND	x/ℓ of fuselage polynomial endpoints; input item 7
FCOEF	coefficients of fuselage polynomials; input item 8

COMMON /SSHOCK/ NSSHK(7),SXSHK(50,7),SRSHK(50,7),SDRDX(101,7)

NSSHK(J)	number of x and r values in Jth store nose shock table generated from line singularities
SXSHK(I,J), SRSHK(I,J)	x and r coordinates of Ith circular store shock location at zero angle of attack for Jth store
SDRDX(K,J)	DR/DX at Kth circular store control point of Jth store

COMMON /SSOR/ SXL(101,7),SSS(100,7),SDS(100,7),NSSOR(7)

SXL(I,J)	Ith x-position of Jth circular store sources; positive measured from tip of nose
SSS(I,J)	Ith source strength of distribution for Jth circular store
SDS(I,J)	Ith doublet strength of distribution for Jth circular store
NSSOR(J)	number of Jth circular store sources and doublets; equal to input item 56

COMMON /STGEOM/ SRMAX(7),SLTHC(7),XWSOC(7),YWSO(7),ZWSO(7),
NSPOLY(7),SXEND(7,7),SCOEF(7,7,7),XBSO(7),YBSO(7),
ZBSO(7),SIBCR(7),NUMSTR(7),SSIBCR(7),CSIBCR(7),
SPHIRR(7),NSHAPE(7),NSHPT,MSHAPE(7)

SRMAX(J)	maximum radius of Jth store; input item 54
SLTHC(J)	length of Jth store; input item 54
XWSOC(J)	x_w coordinate of tip of nose of Jth store
YWSO(J)	y_w coordinate of tip of nose of Jth store

ZWSO(J)	z_W coordinate of tip of nose of Jth store
NSPOLY(J)	number of polynomials used to specify the shape of the Jth store
SXEND(I,J)	endpoints of polynomial sections specifying shape of Jth store
SCOEF(I,K,J)	Jth coefficient of Ith polynomial describing the shape of the Jth store
XBSO(J)	x_B coordinate of tip of nose of Jth store
YBSO(J)	y_B coordinate of centerline of Jth store
ZBSO(J)	z_B coordinate of tip of nose of Jth store
SIBCR(J)	SIC(J)*DTOR; input item 54 for SIC(J)
NUMSTR(J)	store number; input item 54
SSIBCR(J)	sin(SIBCR(J))
CSIBCR(J)	cos(SIBCR(J))
SPHIRR(J)	initial roll angle of Jth store coordinate axes; positive right wing down, radians
NSHAPE(J)	store shape number, input item 54
NSHPT	number of different store shapes; input item 55
MSHAPE(K)	shape number of Kth store shape; input item 56

COMMON /SRCE/	XFIELD,RFIELD,BSQ,X2,U,V,VT
XFIELD	center of line source/doublet segment, feet
RFIELD	circular body radius at XFIELD, feet
BSQ	not used
X2	not used
U	axial velocity due to a source
V	radial velocity due to a source
VT	not used

```
COMMON /THKDAT/ NCWS,NCPS,MS,MPS,MSWS,MSPS,NTHP,XRFT(400),XRBT(400),
     XLFT(400),XLBT(400),YRCT(400),YLCT(400),ZRCT(400),ZLCT(400),
     THETAL(400),THETPL(200),SLLET(400),SLTET(400),DZDX(400),
     YS(20),PSWSLE(20),PSWSTE(20),PHIS(20),ZS(20),PSPSLE(20),
     PSPSTE(20),SNPHS(400),CSPHS(400)
```

NCWS	number of wing thickness panels in a chordwise row; input item 40
NCPS	number of pylon thickness panels in a chordwise row; input item 46
MS	number of thickness panels on wing
MPS	number of thickness panels on pylon
MSWS	number of wing thickness panels in a spanwise row; input item 40
MSPS	number of pylon thickness panels in a spanwise row; input item 46
NTHP	number of thickness panels on wing and pylon
XRFT(I)	x_w coordinate of right front corner of Ith thickness panel
XRBT(I)	x_w coordinate of right rear corner of Ith thickness panel
XLFT(I)	x_w coordinate of left front corner of Ith thickness panel
XLBT(I)	x_w coordinate of left rear corner of Ith thickness panel
YRCT(I)	y_w coordinate of right side of Ith thickness panel
YLCT(I)	y_w coordinate of left side of Ith thickness panel
ZRCT(I)	z_w coordinate of right side of Ith thickness panel
ZLCT(I)	z_w coordinate of left side of Ith thickness panel
THETAL(I)	wing thickness slopes; input item 42
THETPL(I)	pylon thickness slopes; input item 48
SLLET(I)	slope of leading edge of Ith thickness panel
SLTET(I)	slope of trailing edge of Ith thickness panel

DZDX(I)	dz/dx of Ith thickness panel
YS(J)	y_w locations of wing thickness panel side edges; input item 41
PSWSLE(J)	leading-edge sweeps of wing thickness panels; input item 41
PSWSTE(J)	trailing-edge sweeps of wing thickness panels; input item 41
PHIS(J)	wing thickness panel dihedrals; input item 41
ZS(J)	z_p locations of pylon thickness panel edges; input item 47
PSPSLE(J)	leading-edge sweeps of pylon thickness panels; input item 47
PSPSTE(J)	trailing-edge sweeps of pylon thickness panels; input item 47
SNPHS(J)	sin(PHIS(J))
CSPHS(J)	cos(PHIS(J))

COMMON /THVARG/ X,YV,ZV,U,VV,WV,EML,PRT*

X,YV,ZV	x,y,z coordinates of field point in thickness panel coordinate system
U,VV,WV	u,v,w velocity components in thickness panel coordinate system
EML	slope of thickness panel leading or trailing edge
PRT	not used

COMMON /VELARG/ X,YV,ZV,U,VV,WV,EM,TLRNC,TIPY,PYPNL*

X,YV,ZV	x,y,z coordinates of field point in u-velocity panel coordinate system
U,VV,WV	u,v,w velocity components in u-velocity panel coordinate system

---

*Variable names may differ from routine to routine but the definitions are unchanged.

EM	slope of u-velocity panel leading or trailing edge
TLRNC	error tolerance
TIPY	exposed wing span
PYPNL	logical pylon indicator

COMMON /WGEOM/ XBWOC,ZBWO,CRW,SLPWLE,SLPWTE,PSIWLE(20),
PSIWTE(20),Y(20),PHID(20),ZDIHED,WICR

XBWOC	x_B coordinate of wing root chord leading edge; input item 34
ZBWO	z_B coordinate of wing root chord leading edge; input item 34
CRW	wing root chord; input item 35
SLPWLE	tan(PSIWLE(I))
SLPWTE	tan(PSIWTE(I))
PSIWLE(I)	leading-edge sweeps of wing u-velocity panels; input item 37
PSIWTE(I)	trailing-edge sweeps of wing u-velocity panels; input item 37
Y(I)	y_W locations of wing u-velocity panel side edges; input item 37
PHID(I)	dihedral angles of wing u-velocity panel sections; input item 37
ZDIHED	logical variable indicating whether or not there is wing dihedral; ZDIHED=TRUE, no dihedral; =FALSE, there is dihedral
WICR	wing incidence angle, radians; computed from WIC, input item 34

COMMON /WPGEOM/ XRF(200),XRB(200),XLF(200),XLB(200),YRC(200),
YLC(200),SCPT(200),YCPT(200),ZCPT(200),SWPPLE(200),
SWPPTE(200),SNPHI(200),CSPHI(200)

XRF(I)	x_W coordinate of right rear corner of Ith constant u-velocity panel
XRB(I)	x_W coordinate of left front corner of Ith constant u-velocity panel

XLF(I)	x_w coordinate of left rear corner of Ith constant u-velocity panel
XLB(I)	x_w coordinate of right front corner of Ith constant u-velocity panel
YRC(I)	y_w coordinate of left edge of Ith constant u-velocity panel
YLC(I)	y_w coordinate of right edge of Ith constant u-velocity panel
XCPT(I)	x_w coordinate of Ith control point
YCPT(I)	y_w coordinate of Ith control point
ZCPT(I)	z_w coordinate of Ith control point
SWPPLE(I)	sweep of leading edge of Ith constant u-velocity panel on wing, fuselage or pylon
SWPPTE(I)	sweep of trailing edge of Ith constant u-velocity panel on wing, fuselage or pylon
SNPHI(I)	sine of dihedral angle of Ith constant u-velocity panel on wing
CSPHI(I)	cosine of dihedral angle of Ith constant u-velocity panel on wing

COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)

FXSHLD	x_B location of circular fuselage shoulder, feet
RXSHLD	x_R location of rack shoulder, feet
SXSHLD(J)	x_S location of Jth circular store shoulder, feet

B-3 Blank Common

The requirement for handling the many large arrays associated with the solution for u-velocity and source panel strengths has necessitated both the use of out of core data handling and storage and the setting aside of a scratch storage area in core to be used for more than one purpose. To handle the latter data requirement blank common has been reserved for all calculations involving large arrays. The program flow of calculations is thus arranged to allow variables to be read from or written to external files as needed. The following describes the Program I sequence of references to external files and which arrays reside in blank common at each point in the program flow. The information residing in each of the external files is described later.

In Program I blank common is used for three purposes: (1) storing the dynamically dimensioned arrays associated with both the noncircular fuselage and the elliptic store; (2) storing the wing-fuselage u-velocity aerodynamic influence coefficients during the solution for panel strengths; and (3) temporary storage of arrays associated with the summation of u-velocity singularity strengths at panel corners. The descriptions which follow focus on the definition of quantities during these phases. The noncircular fuselage and elliptic store arrays are lumped together because they share common code. Only the dimensions of arrays will vary between components. The arrangement of variables in blank common during these calculations may change dynamically as the solution progresses as temporary arrays are required or discarded. The flow chart in Figure A-2 of Appendix A also identifies several points in the program, item numbers, where external files are referenced. The comments which follow are keyed to the usage of blank common at those points wherever possible. The descriptions of the use of blank common during these three phases follow.

The first use of blank common is to store the arrays of panel properties and solutions of either the noncircular fuselage or the

elliptic store. This spans items 1 through 4 for the fuselage and repeats as item 5 for the store in Figure A-2. The first routine using blank common is GEOM as called from WDYBDY. The noncircular and elliptic store both share the same analysis routines and blank common data arrays. In Program I, the results of the first are computed and all information required for Program II are saved. The solution for a single body is broken into four parts. These follow the progress of calculations and correspond to generation of panel geometry, computation of panel aerodynamic influence coefficients, solution of the equations, and saving of appropriate arrays required for Program II. Because of the large number of possible combinations of paneling schemes, all array storage allocation in blank common is performed dynamically. That is, the array length is computed and core locations set aside prior to calculation of the array itself. Arrays are typically stacked one behind another with no unused locations in between. No explicit array names or dimensions are defined in blank common. Only the address in blank common of the first element in the array is saved in labeled common, DIMENS.

The first use of blank common within the calculation of the source panel geometry (item 1) is for temporary storage of the arbitrary Y,Z input of the external fuselage shape as read directly from input cards. They are followed by the intermediate YB and ZB values computed at axial stations, XFUS, at the new meridional angles, PHIK, of the revised panel spacing layout. Blank common is equivalently dimensioned to:

```
COMMON    SFUS(NRAD,2,NFUSOR),YB(NFUSOR,KRAD),ZB(NFUSOR,KRAD),...
          A(NADIM)
```

where

SFUS(NN,1,N) y-station in local body coordinates of the NNth meridian at the Nth axial station, XFUS

SFUS(NN,2,N) z-station in local body coordinates of the NNth meridian at the Nth axial station, XFUS

YB(N,K) y-station in local body coordinates of the Nth axial station for the Kth revised meridional angle

ZB(N,K) z-station in local body coordinates of the Nth axial station for the Kth revised meridional angle

If more than one segment exists YB and ZB are saved on TAPE8 for each of NFUS segments.

The second change in the use of blank common as identified by item 2 in Figure A-3 is to store the arrays containing the geometric properties of the source panels. The configuration of blank common at this point in the calculations remains the same while continuing to grow. Arrays which are to be saved are located at the beginning with scratch space at the rear. From the end of the calculations in GEOM, the arrays in blank common contain all the panel geometric properties required for calculation of influence coefficients and the resolution of forces and moments. Additional temporary arrays are allocated to hold the ring-by-ring coefficient arrays in VELCMP. The equivalently dimensioned arrays in blank common look like:

```
COMMON   XPT(NBODY),YPT(NBODY),ZPT(NBODY),THET(NBODY),
         DELTA(NBODY),AREA(NBODY),XC(KX),YC(KXKR),
         ZC(KXKR),AN(MAXKR,MAXKR),UB(3,MAXKR,MAXKR),...,
         A(NADIM)
```

where

XPT(I) x-station of control point of Ith panel

YPT(I) y-station of control point of Ith panel

ZPT(I) z-station of control point of Ith panel

THET(I) inclination angle at Ith panel control point

DELTA(I) incidence angle of Ith panel

AREA(I) surface area of Ith panel

XC(L,M) panel corner points at Lth axial station of Mth segment

YC(L,N,M)	y corner point at Lth axial station of Nth meridional angle of Mth segment
ZC(L,N,M)	z corner point at Lth axial station of Nth meridional angle of Mth segment
AN(I,J)	temporary array for aerodynamic influence coefficients
UB(1,I,J)	temporary u-component influence coefficient array
UB(2,I,J)	temporary v-component influence coefficient array
UB(3,I,J)	temporary w-component influence coefficient array

The dimensions and indices defining the starting locations of each of these equivalent arrays are found in common DIMENS. The starting indices may typically be found by preceding the first two or three letters of the array name with I.

Additional arrays associated with the solution for the panel source strengths in SOLVE as identified by item 3 in Figure A-2. Only the arrays containing the panel strengths are saved. The strengths for up to seven flow conditions may be retained to allow for using the same shape data for more than one store. The equivalently dimensioned arrays in blank common looks like:

```
COMMON   XPT(NBODY),YPT(NBODY),ZPT(NBODY),THET(NBODY),
         DELTA(NBODY),AREA(NBODY),XC(KX),YC(KXKR),ZC(KXKR),
         GB(NBODY,IALP),VB(NBODY),AN(MAXKR,MAXKR),...,
         A(NADIM)
```

where the arrays XPT through ZC contain the same information as before and

GB(I,J)	strength of Ith panel for fuselage (J=1) or Jth store of given shape
VB(I)	temporary array; contains initial velocity boundary condition which is destroyed during the solution

AN(I,J) temporary array; used to hold the aerodynamic influence coefficients

At the conclusion of the calculation of the fuselage body source strengths identified by a call to IOWRIT in Figure A-2, arrays XPT through GB are saved on TAPE7. At the conclusion of the remaining body calculations all arrays required for Program II are saved on TAPE10. This occurs during the call to routine FRSTRT identified as item 4 in Figure A-2. For the fuselage, all variables and arrays in common blocks BGEOM, BOPTNS, BINLET, BINSHK, BSHOCK, DIMENS, HEAD, PARAM, and the blank common arrays previously saved on TAPE7 are written onto TAPE10. These commons contain all the information necessary to restart the calculations in Program II.

In Program I when both a noncircular fuselage and an elliptic store are to be analyzed, the results of the fuselage calculation are saved on an external file as indicated above. The data for the store or stores overwrite previous values. The sequence employed in computing elliptic store panel strengths is performed in a slightly different manner from the fuselage. In order to take maximum advantage of similarities in store shapes, the geometric properties of the panels and aerodynamic influence coefficients of a given shape are computed only once. The placement of a store in a different position or orientation causes only a variation in the boundary conditions. Only the strengths of the source panels associated with the additional stores of identical shape have to be saved. Provision for two elliptic store shapes is included in the program.

The computation sequence for the stores has been modified to read the geometric store input by shape first. The aerodynamic influence calculations are then made for each individual store. The impact of this sequence of computations for the store in blank common is to break the sequence 1 through 4 as used for the fuselage after item 2 in order to temporarily save all arrays

on TAPE11. This break in the sequence is indicated by item 5 in Figure A-2. At this point arrays XPT through ZC and common blocks BGEOM, BOPTNS, BSHOCK, DIMENS, HEAD, and PARAM are saved for each of the elliptic store shapes as previously described for the fuselage. The loop on the number of shapes is then repeated in which the above variables are read and the source strengths computed for each of the stores of that shape. The above labeled common blocks and arrays XPT through GB are saved on TAPE10 by routine FRSTRT. Blank common is then used to copy the aerodynamic influence coefficients from TAPE9 and TAPE8 to TAPE10. Blank common is equivalently dimensioned as

```
COMMON   AN(KRAD,KRAD),...,A(NADIM)
```

for the TAPE9 transfer and then

```
COMMON   UB(3,KRAD,KRAD),...,A(NADIM)
```

for the TAPE8 transfer of data.

TAPE10 is rewound and the records containing the fuselage arrays are read for use by DPHRS at item 6 in Figure A-2. FRSTRT reinitializes blank common and the various labeled commons identical to that configuration at the point identified by item 3 to allow use of the noncircular fuselage solution for the wing boundary conditions.

The second use of blank common in Program I is to contain the aerodynamic influence coefficients associated with the u-velocity panel solution for the wing-fuselage-pylon of the parent aircraft. At item 7 in Figure A-2 DPCOEF uses blank common to hold the coefficients and the right hand side through the panel strength solution. For these computations blank common has the dimensions of

```
COMMON   FVN(200,N+1)
```

where

FVN(I,J) u-velocity panel influence coefficients

FVN(I,N+1) on input is the boundary condition of the Ith panel; on output is the strength of the Ith panel

N number of u-velocity panels

The third use of blank common in Program I is to contain the arrays used in the condensation of u-velocity panel strengths to strengths associated with single corners. Blank common is used to contain the arrays of corner points, net strengths, and orientation information. These calculations are performed in routine NULYT identified as item 8 in Figure A-2. The information is retained in blank common only until it can be written onto TAPE12 by WRFILE. During this time blank common has the dimensions of

```
COMMON    XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
          SWP(500),SNBP(40),CSBP(40),THTBP(40),DPNET(500),
          XPTS(1000),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
          SWPS(1000),THTNET(1000)
```

where

XPT(I) x_w coordinate of Ith wing/body/pylon constant u-velocity corner point

YPT(J) y_w coordinate of Jth chordwise row associated with wing/body/pylon constant u-velocity corner points

ZPT(J) z_w coordinate of Jth chordwise row associated with wing/body/pylon constant u-velocity corner points

SPHI(J) sine of dihedral angle associated with Jth row of wing constant u-velocity corner points

CPHI(J) cosine of dihedral angle associated with Jth row of wing constant u-velocity corner points

SWP(J) leading-edge slope of semi-infinite influencing triangle associated with Jth constant u-velocity corner point

SNBP(J)	sine of orientation angle of Jth row of body interference shell u-velocity corner points
CSBP(J)	cosine of orientation angle of Jth row of body interference shell u-velocity corner points
THTBP(J)	polar angle in cross-sectional plane defining the Ith row of fuselage constant u-velocity corner points; positive in counterclockwise rotation from positive y_B axis
DPNET(I)	net strength of Ith constant u-velocity corner point
XPTS(I)	x_W coordinate of Ith wing/pylon thickness source panel corner point
YPTS(J)	y_W coordinate of Jth chordwise row of wing/pylon thickness source panel corner points
ZPTS(J)	z_W coordinate of Jth chordwise row of wing/pylon thickness source panel corner points
SPHS(J)	sine of dihedral angle associated with Jth row of wing source panel corner points
CPHS(J)	cosine of dihedral angle associated with Jth row of wing source panel corner points
SWPS(I)	leading-edge slope of semi-infinite influencing triangle associated with Ith source panel corner point
THTNET(I)	net strength of Ith thickness source panel corner point

Blank common is last used in Program I by FRSTRT as previously mentioned as temporary array space to copy the source panel data for the fuselage and store bodies from TAPE10 to TAPE12. See descriptions of FRSTRT for sequence of operations involved in file transfer. This reference to blank common is identified as item 9 in Figure A-2.

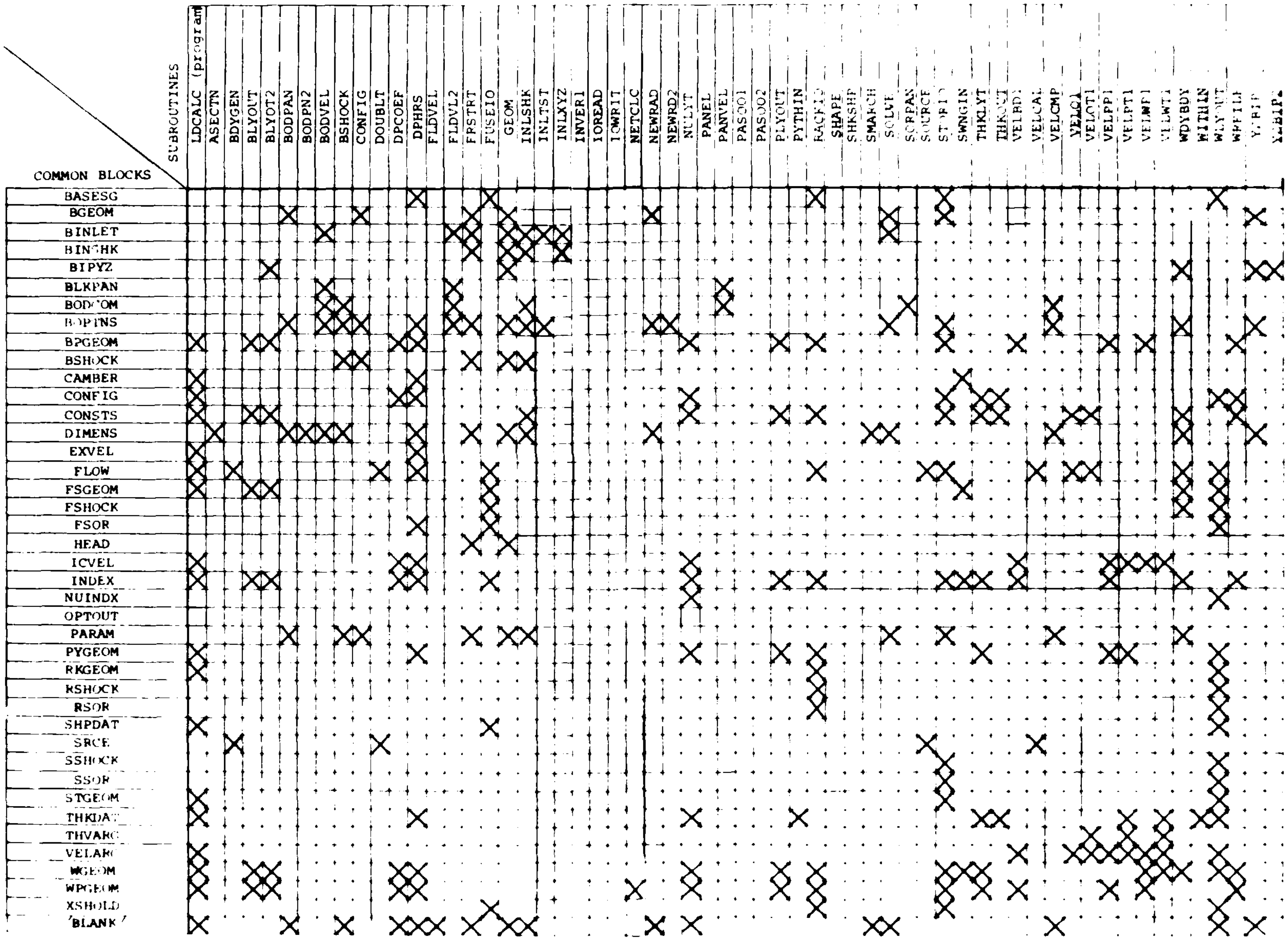


Figure B-1.- Common statements and the routines in Program I in which they appear.

REFERENCES

1. Dillenius, M. F. E., Goodwin, F. K., and Nielsen, J. N.: Prediction of Supersonic Store Separation Characteristics. Vol. I - Theoretical Methods and Comparisons with Experiment. Technical Report AFFDL-TR-76-41, Vol. I, May 1976.

2. Krasnov, N. F.: Aerodynamics of Bodies of Revolution. Edited by Deane N. Morris, Elsevier, New York, 1970.

3. Goodwin, F. K., Dillenius, M. F. E., and Mullen, J., Jr.: Prediction of Supersonic Store Separation Characteristics Including Fuselage and Stores of Noncircular Cross Section. Vol. I - Theoretical Methods and Comparisons with Experiment. Technical Report AFWAL-TR-80-3032, Vol. I, Nov. 1980.

4. Woodward, F. A.: An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow, Part I - Theory and Applications. NASA CR-2228 Part I, May 1973.

☆U.S.Government Printing Office: 1981 · 757-002/461